

LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A241 039



DTIC ACCESSION NUMBER

LEVEL

1

INVENTORY

WL-TR-91-8045

DOCUMENT IDENTIFICATION

Aug 91

DISTRIBUTION STATEMENT A
Approved for public release
Distribution is unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR

NTIS GRA&I
DTIC TRAC
UNANNOUNCED
JUSTIFICATION

☒ ☐ ☐

BY

DISTRIBUTION/

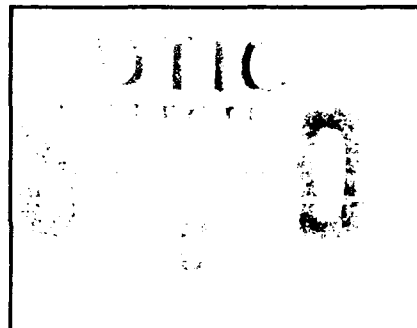
AVAILABILITY CODES

DISTRIBUTION

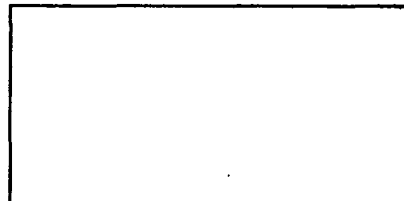
AVAILABILITY AND/OR SPECIAL

A-1

DISTRIBUTION STAMP



DATE ACCESSIONED



DATE RETURNED

91-11971



DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

H
A
N
D
L
E

W
I
T
H

C
A
R
E

AD-A241 039



UNIFIED LIFE CYCLE ENGINEERING (ULCE) DESIGN SYSTEM

Patrick R. Flanagan

KDT Industries
1301 W. 25th Street, Ste 506
Austin, Tx 78705

August 1991

Final Report for period February 1990 - August 1990

Approved for public release; distribution is unlimited.

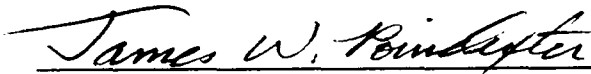
MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

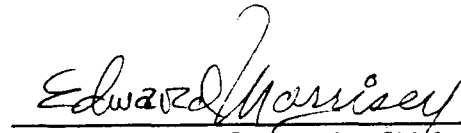
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

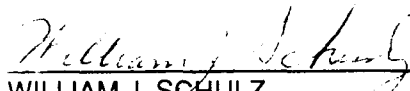
This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.


JAMES W. POINDEXTER, Mechanical Engr
Concurrent Engineering Office
Manufacturing Technology Directorate


EDWARD MORRISEY, Acting Chief
Concurrent Engineering Office
Manufacturing Technology Directorate

FOR THE COMMANDER


WILLIAM J. SCHULZ
Acting Director
Manufacturing Technology Directorate

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/MTR, WPAFB, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			5. MONITORING ORGANIZATION REPORT NUMBER(S) WL-TR-91-8045	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			7a. NAME OF MONITORING ORGANIZATION Manufacturing & Technology Directorate (WL/MTR) Wright Laboratory	
6a. NAME OF PERFORMING ORGANIZATION KDT Industries, Inc.		6b. OFFICE SYMBOL (if applicable)	7b. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433-6533	
6c. ADDRESS (City, State, and ZIP Code) 1301 W. 25th Street, Ste 506 Austin, Tx 78705		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-89-C-5733		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (if applicable)	10. SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code)		PROGRAM ELEMENT NO. 65502F		PROJECT NO. 3005
		TASK NO. 52		WORK UNIT ACCESSION NO. 61
11. TITLE (Include Security Classification) UNIFIED LIFE CYCLE ENGINEERING (ULCE) DESIGN SYSTEM				
12. PERSONAL AUTHOR(S) Patrick R. Flanagan				
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM 2/90 TO 8/90		14. DATE OF REPORT (Year, Month, Day) August 1991
15. PAGE COUNT 76				
16. SUPPLEMENTARY NOTATION This is a Small Business Innovative Research Program, Phase I report.				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	Feature Based Modeling	
			Life Cycle Analysis	
			Concurrent Engineering	
			Simultaneous Engineering	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)				
<p>This report describes Phase I of a project to develop a computer program to improve the efficiency of the engineering design process. The program will integrate Feature Based Modeling with Concurrent Engineering to facilitate the development of life cycle analysis models. The purpose of the program is to reduce data management and computer program interface coding requirements on engineering designers. The proposed program combines an Object Oriented Database, Expert System, Computer Aided Design (CAD) system, and computer model integration tool to create friendly, graphically oriented user interfaces that allow multiple representations of design data. This program also allows designers to develop interfaces between the design database and computer analysis models without writing computer code. The project resulted in four primary products: 1) a methodology and architecture, 2) a limited proof of concept computer model, 3) a feature based life cycle model of a microwave switch, and 4) a Phase II plan.</p>				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL James W. Poindexter			22b. TELEPHONE (Include Area Code) (513) 255-8589	22c. OFFICE SYMBOL WL/MTR

TABLE OF CONTENTS

Purpose of the Work	1
Concurrent Engineering.....	1
Cost Effectiveness.....	3
Product Improvements.....	3
Process Improvements.....	4
Development Time.....	6
Complexity.....	7
Feature Based Modeling.....	7
Difficulties of Concurrent FBM.....	9
Description of the Work	12
Examine the State of the Art.....	12
Tools.....	12
Research.....	13
Define A Solution.....	14
Power.....	14
Friendliness.....	15
Description of the Proposed System.....	16
Proof-of-Concept System.....	23
Phase II Plan.....	24
Results	25
Proof-of-Concept Program.....	25
Layers.....	25
Analysis.....	29
Advice.....	30
Microwave Switch Application.....	30
Potential Applications	33
Bibliography	34

LIST OF FIGURES

Figure 1.	Concurrent Engineering.....	2
Figure 2.	Sequential Engineering.....	2
Figure 3.	Ability to Influence Final Cost on Project Life	5
Figure 4.	Engineering Design Process.....	17
Figure 5.	Example Assembly List Frame.....	18
Figure 6.	Alternative Data Views	21
Figure 7.	Proof-of-Concept Bill of Materials	26
Figure 8.	Example Bill of Materials Layer.....	28
Figure 9.	Model Execution Order.....	29
Figure 10.	Advice from the Proof-of-Concept Program	31

LIST OF TABLES

Table 1.	Life Cycle Processes	6
Table 2.	Knowledge Required for Processes	10

PURPOSE OF THE WORK

Phase I of this project provided the foundation for the development of a computer program to integrate Concurrent Engineering (CE) with Feature Based Modeling (FBM).

The project has been motivated by the enormous potential for improving the design process and by the difficulty of implementing concurrent feature based design. Concurrent Engineering and Feature Based Modeling are both attempts to deal with the complexities of design by organizing the process. Neither differs from traditional good engineering practices. The difficulties of:

- storing and accessing important data,
- integrating tools that do parts of design analyses, and
- manipulating outputs to understand the ramifications of the results,

reduce the designer's ability to be creative. Simplifying the design data handling and code generation burdens on the design engineer will foster creativity and is the focus of the project.

Feature Based Modeling facilitates Concurrent Engineering by providing a common store of data relevant to design analysis. Together, FBM and CE hold the promise of better, more cost effective designs. By considering all life cycle phases early in design, cost effectiveness over the entire life can be improved. Rather than developing designs that are good locally for the design or manufacturing phases and difficult to operate and support, good global solutions can be developed that offer low life cycle costs.

CONCURRENT ENGINEERING

Concurrent Engineering is also known as Simultaneous Engineering. CE begins during the conceptual design phase and concurrently considers the cost effectiveness of an object during all of the processes it will undergo in its life. Figures 1 and 2 show the differences between Concurrent and traditional, Sequential Engineering. CE requires the development of models of all of processes of the life of the object as a part of design.

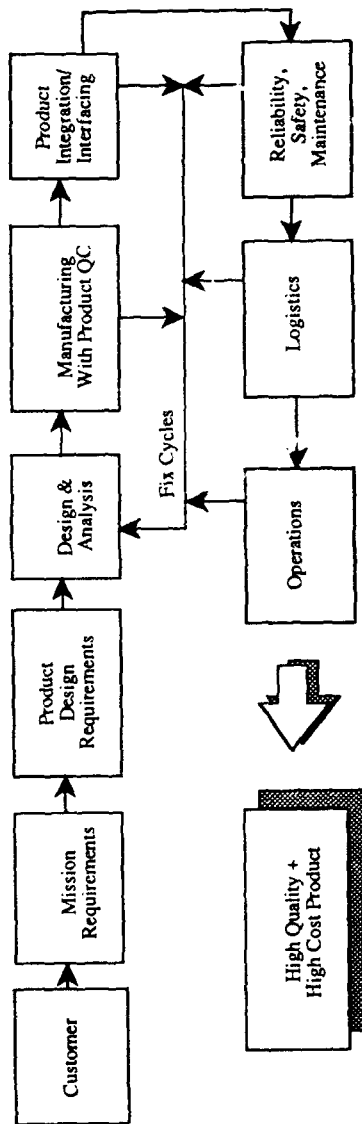


Figure 1 - Traditional Sequential Approach

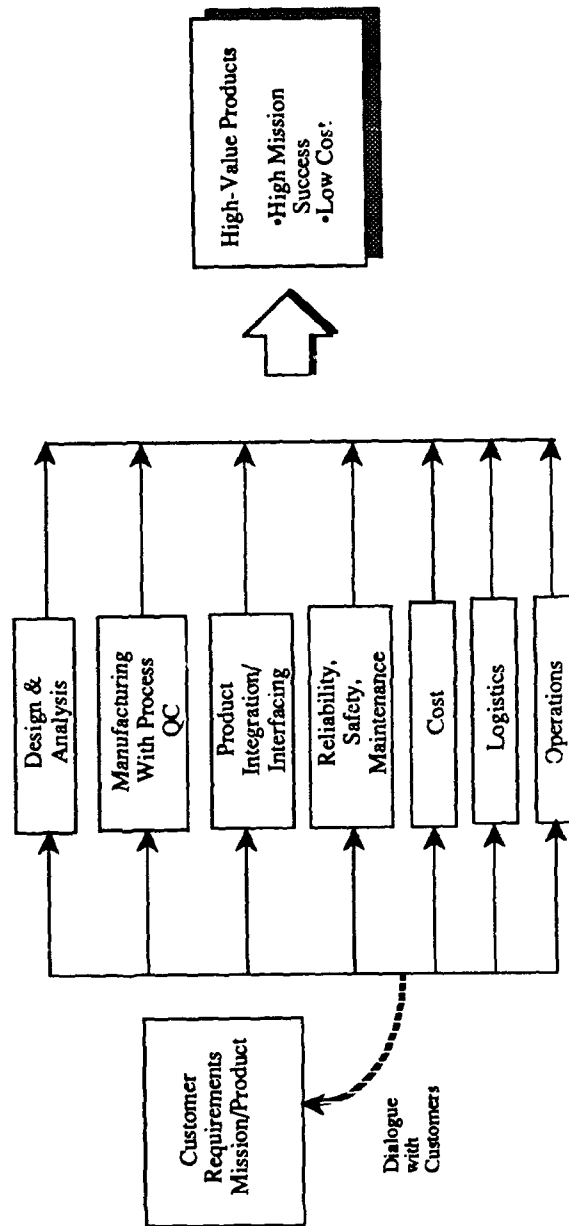


Figure 2 - Concurrent Approach

The growing interest in concurrent engineering is being driven by the need to:

- improve cost effectiveness of designs,
- reduce development times, and
- deal with design complexity.

Cost Effectiveness

Effectiveness is the measure of the ability of a system to achieve a set of required performance parameters such as: fuel economy, speed, probability of detection, probability of kill, ready rate, and cost. Cost is the cost of designing, producing, operating, and disposing of the system. Cost effectiveness is, therefore, the cost of achieving performance throughout the life of the system. Life cycle cost effectiveness is an important measure because much of the total cost of a system occurs during operation and disposal phases. Traditionally costs have not been considered during the procurement process, and while production costs are important for determining the life cycle cost effectiveness, they may be greatly overshadowed by support costs.

Another reason for performing concurrent engineering is to improve designs. Design improvement is accomplished by improving both the product itself and the processes it undergoes. Some examples of product improvement are: quicker response, lower cost, improved accuracy, and longer Mean Time Between Failures (MTBF). Examples of process improvements include: fewer and quicker setups during manufacturing, fewer tasks during maintenance and testing, and fewer tools required for replacement and testing. Product and process improvements are closely linked. For example, the repair process for a system may be improved by decreasing repair time as a result of designing the product with fewer parts.

Product Improvements

Better product designs are achieved by considering more phases of the life cycle. For example, human factors for the maintenance and support phases of operation are often considered as an afterthought. Parts are sometimes designed that cannot be easily serviced because access for service was not considered during the design phase. By involving more disciplines early in the design process, designers can avoid some of the pitfalls of producing designs that cannot be supported or maintained. Stakeholders from all phases

must be considered early in the design process when their input can be cost effectively considered. Much of the cost of a design is committed long before it is spent. Figure 3 shows an example of the degree to which change can be affected in a design with time. Design modifications are much easier to make early in the design process than they are later.

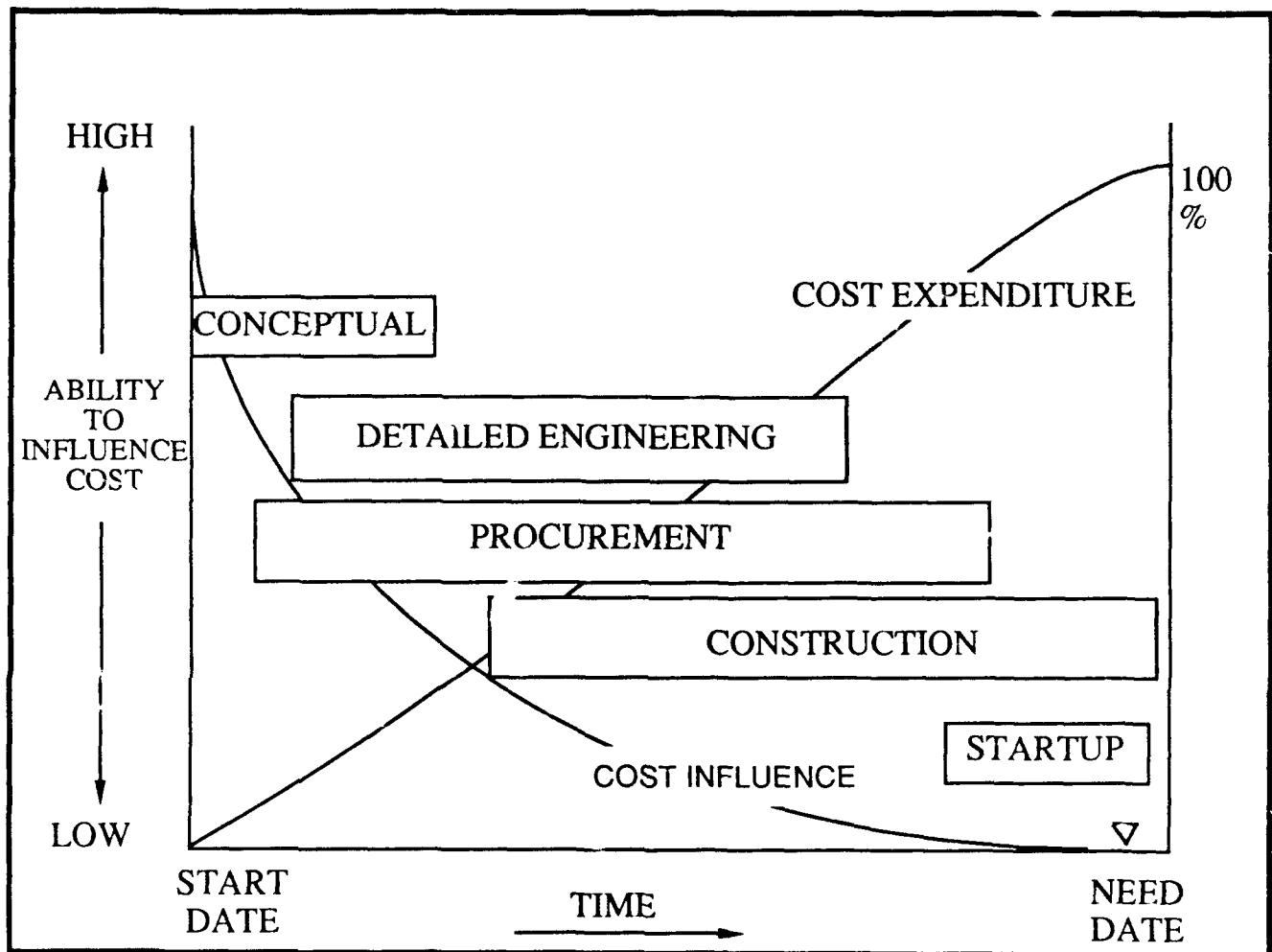
Considering the life cycle of a system early can reduce the cost of design and redesign in several ways. As an example, the wiring for the advanced F-16 bomb rack being proposed for Phase II of this project will be designed with the ability to accept proposed Military Standard interfaces. Although the standards have not been finalized, the design will allow the advanced bomb racks to be upgraded easily. Upgrades will only be necessary for those aircraft and missions that require upgraded capabilities. The proposed design will reduce maintenance, modification time, and spares level required when the new interface is accepted. By considering this later phase in the bomb rack life cycle up front, redesign and modification costs will be reduced.

Concurrent engineering can also reduce the costs of redesign by developing phase interfaces early. The ways in which the different phases of system life impact each other is more likely to be considered in concurrent engineering, as these processes are considered early. By designing the way the system operates in all phases and by designing the processes that support this operation, the likelihood is reduced that a system must be redesigned because it cannot either be operated efficiently or supported well.

Considering all phases early can allow the designer to find a good global solution, improving cost effectiveness over the entire life. The designer may move away from a locally good solution to improve overall cost effectiveness. For example, the designer may specify a set of manufacturing tolerances that reduces total cost by either lowering manufacturing costs and reducing effectiveness or improving performance through tighter tolerances. It may even be possible to both reduce manufacturing costs and improve performance.

Process Improvements

While the need for an improved product appears self evident, the reasons for improving the processes that involve the product are not so obvious. Within the last decade, it has become clear that process improvements are at least as important as product improvements.



Reprinted from "Special Supplement 1987 Transactions of the American Association of Cost Engineers"

Figure 3 - Ability to Influence Final Cost Over Project Life

Processes are often more important in terms of cost, technical feasibility, and the ability of the product to meet performance specifications than are the characteristics of the product itself. Life cycle processes are those activities that affect a design during its life. Table 1 provides some examples of life cycle processes.

Table 1 - Life Cycle Processes

Phase	Processes
conceptual design	scope, brainstorming, storyboarding;
specification	CAD drawings, bill of materials, route sheets, assembly procedures; prototype model, stereo lithography;
testing	test specification, data collection, data synthesis and analysis;
manufacturing	small scale production, first article full production;
operation	training, operating procedures, consumables support, maintenance, repair, logistics;
disposal	training, site preparation procedures, and monitoring activities

In addition, poorly designed processes can reduce the effectiveness with which the system operates. The technical feasibility of systems may be limited by the feasibility of the processes supporting it. As an example, questions have been raised about the effectiveness of Space Station Freedom given large estimates for extravehicular activities required to assemble the station. The number of hours that are required before the station can be occupied may not be feasible.

Processes may also limit the performance characteristics of the product. For example, excessive training requirements for system operation may make the system difficult to use effectively.

Development Time

The parallel development process of Concurrent Engineering translates into reduced development time requirements. Because all phases of the life cycle are designed early in the process, they can be put in place in parallel. Not only are processes ready sooner, there is more time to fine tune the processes for improved efficiency. As an example, automobile designs are often given to manufacturing engineers eight months before production is scheduled to commence. This is insufficient time to produce an efficient manufacturing

operation. The results are delays or a production process that continues to be developed while production is occurring.

The possibility of design errors is reduced by designing for the entire life cycle. By designing the manufacturing process while the item itself is being designed, there is a reduced likelihood of creating designs that cannot be manufactured. In addition, these kinds of design problems can be discovered early when they are less expensive to correct.

As a result of parallel development and the reduced possibility of error, development lead time can be reduced by concurrent engineering.

Complexity

The size and complexity of development projects also motivates the adoption of Concurrent Engineering. The side effects of design decisions and design changes can overshadow the direct effects due the complexity of design process. By adopting Concurrent Engineering techniques, interactions between the product and processes can be explicitly defined early, reducing the need for design changes and reducing the effects of changes that do take place.

Design revisions can result in large costs to a project, due to both the direct impact costs and the costs due to delay and disruption. If a design change must be made late in the project development, the so-called "ripple effects" are far more numerous and costly than those caused by a design change made in the conceptual development phase. In 1978, the Navy settled out of court with Litton Industries as a result of proof that Navy-responsible delays and design changes had resulted in a \$447 million cost overrun to a firm-fixed-price contract. (Cooper, 1980).

FEATURE BASED MODELING

Definition of supporting processes is as important for Feature Based Modeling as it is for Concurrent Engineering. Feature Based Modeling defines objects in terms of the processes that are important for the object. These object descriptions are in the form of descriptions of features the object possesses. Features are used in analyses of the design to determine

its merit. Therefore, a feature can be any description of the object that makes a difference in the object's value. Some common features are:

- dimensions,
- material,
- weight,
- surface finish,
- manufacturing tasks, and
- geometry.

Each of these features is defined in terms of some process. For example, dimensions, material, surface finish, and manufacturing tasks all help define the manufacturing process used to produce the object. Weight, surface finish, and geometry will effect the operation of the object, while dimensions and weight will effect servicing requirements.

FBM differs from traditional modeling because the features are explicitly defined, rather than inferred, from drawings, bills of material, or experience. FBM features are also defined in terms of the processes they describe and the analyses in which they will be used. For example, a hole may be described as a drilling operation for manufacturing.

FBM is more closely linked with geometric descriptions of designs than are traditional Concurrent Engineering applications. CE applications have been concerned primarily with more abstract issues because the of the difficulty of integrating geometric data with other data types in one system.

The importance of FBM is that the features that describe the design can be used to drive computer analysis models of the design. The features are explicit, unambiguous descriptions of the design that also describe its supporting processes. The link with graphical descriptions of objects is also an important part of FBM. People think graphically and drawings are an efficient means of transmitting information about an object.

Design is traditionally done by drawing rather than by creating textual descriptions. Many important features that describe an object are related to its geometry, such as:

- shape,
- volume,
- assembly process, and
- interfaces.

Feature Based Modeling facilitates design analysis by creating a single, well-defined model of the object. FBM allows integrated design analyses rather than a number of disjointed analyses using data from Computer Aided Design (CAD) drawings, Work Breakdown Structures (WBSs), parts lists, and manufacturing route sheets.

DIFFICULTIES OF CONCURRENT FEATURE BASED MODELING

While integrating CE and FBM holds great promise for improving engineering designs and decreasing the time required to create those designs, performing concurrent feature based modeling is difficult for a number of reasons:

- volume of data,
- disjoint types of data,
- breadth of knowledge required, and
- number of analyses required.

The volume of data required for adequately modeling the life cycle of a real system is difficult to store, manipulate, and view. Data collection, data storage, and the ability to access and manipulate data are all difficult problems when considering the volumes of data required.

The number of data types that are important for modeling a system are beyond the abilities of traditional database management systems. A system that integrates FBM with CE will require the ability to store and access data types typical to traditional databases as well as graphical data and information about influences that data have on each other.

The amount of knowledge needed to develop good life cycle designs is also a deterrent to concurrent feature based modeling. The design team must understand the impacts of

design decisions within each phase of the life as well as across phases. For example, reducing weight to improve operational performance may require that materials be used that are difficult to manufacture. The knowledge required of a design team includes information about the product and its supporting processes such as:

Table 2 - Knowledge Required for Designing Processes

design process	conceptual design storyboarding drawing breadboarding prototype pilot plant
manufacturing process	machining operations material properties machine availability
operations process	mission environment staffing numbers and skills training human factors
support	staffing numbers and skills training equipment/facilities maintenance procedures test procedures replacement repair spares levels requirements positioning policy

The expertise required to develop a model considering each of these requirements is beyond the knowledge of one designer or small group of designers.

An additional deterrent to Concurrent Feature Based Design is the effort required to develop the analytical portion of life cycle analyses. Linking models to analyze the phases of a design requires that interfaces be developed to a number of computer models.

DESCRIPTION OF THE WORK

The work performed in Phase I was to:

- survey the state of the art in Feature Based Modeling and in Concurrent Engineering,
- define a development effort that both addressed the need to integrate FBM and CE and was achievable,
- develop a limited proof of concept computer program to show that the proposed system could achieve its goals, and
- describe a plan for the development of the system in Phase II.

EXAMINE THE STATE OF THE ART

The initial project task was to examine existing tools and techniques that use Feature Based Analysis or Concurrent Engineering to define the state of the art.

Tools

FBM has traditionally been closely related to CAD tools. Until recently, CAD tools have been hampered by the difficulty of storing and accessing non geometric data with drawings. The need to store sufficient additional data to render designs unambiguous for Concurrent Engineering has led to the development of Feature Based Modeling tools.

A number of tools have been developed to facilitate Feature Based Modeling. The Concept Modeller created by Wisdom Systems is an example of a Feature Based Modeling system. The Concept Modeller is used by Wisdom Systems to apply Simultaneous Engineering techniques that have been termed Simultaneous Engineering Automation. Wisdom Systems proposes to apply the Concept Modeller to the solution of design problems from the design process through manufacturing and operations. In addition, traditional CAD packages are being modified to handle more kinds of data. The main drawbacks to using existing tools for Concurrent Engineering is that they may require the use of proprietary CAD or analysis packages or interfaces between the design specification and analysis to be developed in either the LISP or C++ programming languages .

Concurrent Engineering is a field that has received a great deal of interest recently. Special tools that address parts of the Concurrent Engineering process such as Quality Function Deployment (QFD) and Taguchi analysis have been developed and are being marketed. Some systems have been developed to calculate the life cycle costs of software and hardware development efforts based upon regression analysis of historical data. Most other Concurrent Engineering development efforts are proprietary systems not marketed outside the developing organization, or remain research efforts.

Research

Much of the research into Feature Based Modeling examines issues of feature definition and automatic feature extraction. Feature definition research has focused on defining features in terms of their use in analysis programs. An example of this type of research defines geometric shapes in terms of the manufacturing operations required to produce them. For example, a hole could be defined as a drilling or punching operation depending upon the material containing the hole, machine capabilities, and machine availability. Research is underway to define a complete set of machining operations that describe geometric shapes.

Automatic feature extraction is an area characterized by a great deal of research. This research topic has received a large amount of funding due to its application to automatic target acquisition. Although imminent results have been announced for target acquisition, automatic design feature extraction will likely remain a research topic for some time.

Concurrent Engineering research efforts are dominated by the DARPA Initiative for Concurrent Engineering (DICE) and Computer-aided Acquisition and Logistics System (CALS) programs. DICE efforts have been focused by DARPA on developing a computer system that store and transmit the proper information to Concurrent Engineering team members based on their need to know. CALS efforts are directed at developing a standard for passing design information via computer. KDT, Industries, Inc. has contacted and received information from DICE and CALS programs concerning their efforts and will continue to monitor the status of these programs.

DEFINE A SOLUTION

After surveying the state of the art, the scope of several options were evaluated with respect to the solution to be proposed for Phases I and II. KDT Industries, Inc. development efforts have been characterized by trying to help people do what they already do in a more efficient manner, rather than try and create a system that radically changes the way things are done. This approach has been beneficial for two reasons. First, paradigm jumps are relatively rare. Proposals to radically change the way things are done are seldom successful. Second, there is a great deal of resistance to radical changes in the way things are done. People have a natural resistance to trying new solution methods in part because of their investment of time and capital in existing methods.

The complexity of concurrent analyses is driven by: 1) the amount of data to be handled, 2) the number of computer interfaces to analysis programs that must be developed, 3) the difficulty of making sense of the results, and 4) the breadth of knowledge required to make cost effective design decisions over the the entire life cycle.

It was decided to attack the complexity of Concurrent Engineering by developing a system that is both friendly and powerful, reducing the extra burdens of data management and computer interface coding on designers. The purpose of the proposed system is to make data management less taxing and easier to understand, reduce coding requirements for developing life cycle model interfaces, and provide a means of capturing and storing knowledge about designs. None of these objectives is to be accomplished by radical changes in the design process, development of new CAD or analysis programs, or by advances in Artificial Intelligence or Computer Science. The objectives are to be accomplished by developing a computer program focused on providing designers with easy means of inputting, manipulating, and analyzing data.

Power

The power of this system is derived from its ability to access all data from a central Object Oriented Data Base (OODB). This data base will hold either the actual data or references to data locations. An object oriented data base was chosen because of its ability to handle many different kinds of data, including graphical data and references to other data.

An additional source of the power of the system is in the ability of the designer to define features. Because it would be impossible to develop a feature list of sufficient generality to describe all features of all parts of all systems, the system allows the designer to define features by combining primitive data types. These data types include the basic data types traditionally associated with data base managements systems such as character, numeric, logical, and memo type data fields. The system will also provide the designer with the ability to define and access lists, geometric, pointer, coordinate information, and design rules-of-thumb. These data types allow the designer to build feature descriptions that match the system being designed and are important to him.

While the system will allow the designer to store and access expert assistance about the design, the proposed system will not be capable of automatic design or learning. These are topics of advanced Artificial Intelligence (AI) research and not appropriate for a commercial system. The purpose of the proposed expert assistance feature is to give the designer the ability to remember the things that affect the design, store historic data about runs, provide the designer with advice for designing experiments, and help determine the first order isolated effects of changes one at a time. The system will help the user enter rules, set up truth maintenance in which discrepancies are noted but not fixed, and access the rules in a network to allow the user to navigate around the advice set to determine how to change the design.

The system power also stems from the separation of the analysis function from database and assistance functions. The ability to create interfaces between the user's analysis tools allows the user to retain the investment in tools and training. Rather than try and impose an analytical tool set on the designer, the proposed system will allow the user to develop the interfaces to his tools.

Friendliness

The power of the system as described above is certainly no greater than that of a high level computer programming language such as the C programming language. In order to further the state of the art, the system must couple power with friendliness. The friendliness of the proposed system is achieved by allowing designers to enter and view data in ways that make the most sense to them.

Many data representation schemes have been developed for design. These representations include: Work Breakdown Structures, Bills of Material, scheduling relationships, drawings (especially CAD drawings), manufacturing route sheets, and stress/strain diagrams. Designers must be able to generate these representations either through the proposed system or through analysis tools and link them in ways that make sense to them.

In addition to facilitating data management, the system must allow the designer to structure and run analyses without the need to develop computer code. The designer will be able to develop graphical representations of the order in which models are to be run, and specify the interfaces between the models by pointing and clicking with the mouse to define sources and targets for data.

Description of the Proposed System

The proposed system allows the designer to describe a design, analyze the design, and make design changes. This process is shown in Figure 4. The designer develops a database of the important features defining a design. These features are used by analysis models to develop figures of merit for the design. The designer is then able to obtain expert advice for changes in feature descriptions. The revised design is reanalyzed and changed iteratively until the designer is satisfied that the design requirements are met.

The system defined for Phase II development is composed of four parts:

- an object oriented database,
- a CAD program,
- a program to allow easy analysis tool integration, and
- an expert system.

At the center of the system is an object oriented database. The database acts as the central store of information about the features describing the design. It also contains the information about how the features are defined and how they relate to each other.

Features are stored in a data structure called a frame that is created by the designer. A frame is a collection of data defined by the designer. Although the frame structure will depend upon the designer's preferences and the object being designed, some frame

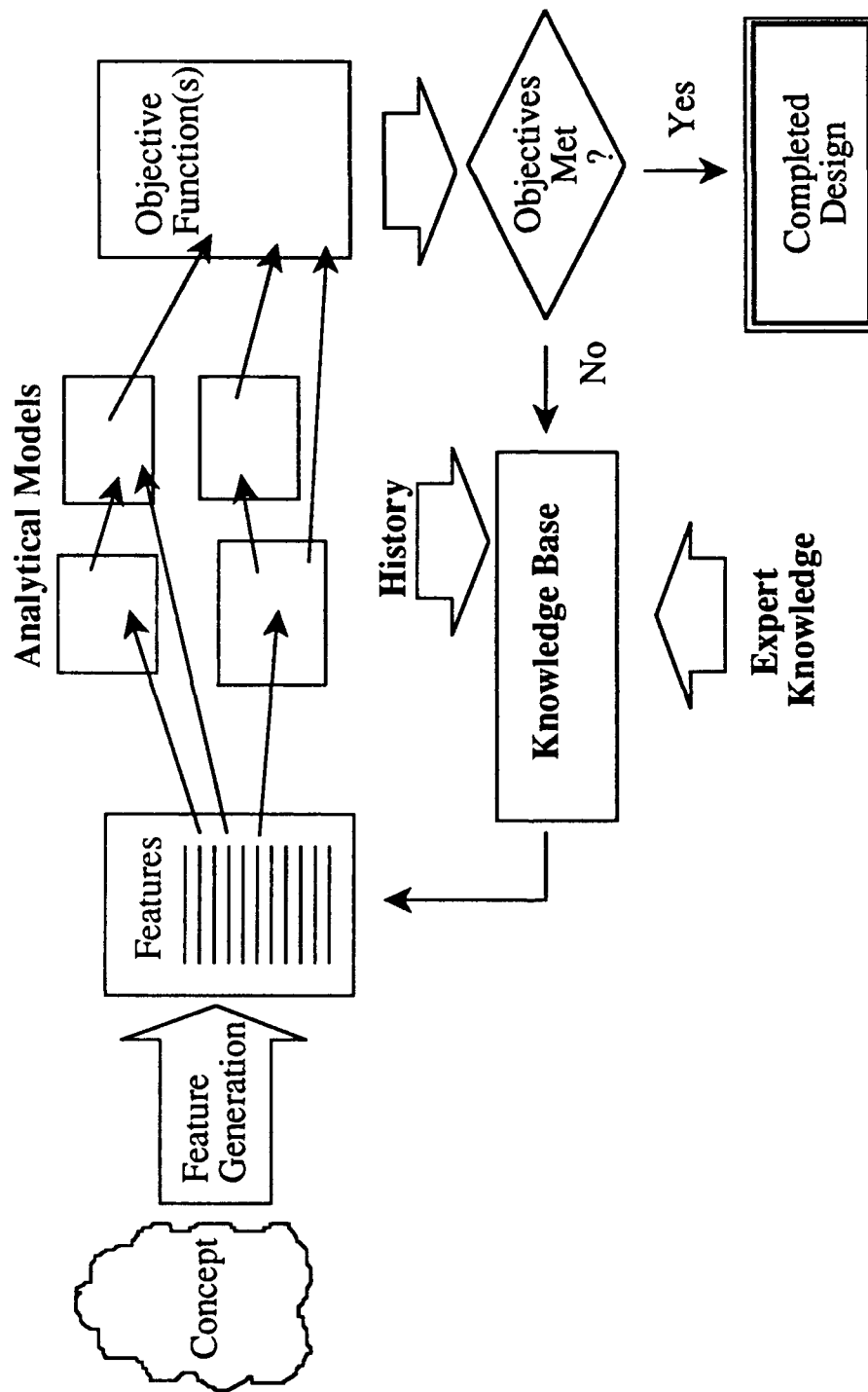


Figure 4 - Engineering Design Process

structures would seem likely to be useful for many model specifications. Figure 5 shows an example frame based on a hierarchical component breakdown of a system into its components.

Number:	1.2.2.2
Name:	Coil
Part Number:	Copper 303-400-445
Component of:	1.2.2 Coil SA
Component List:	-----
Geometry:	<div> <div>part;</div> <div>rotation;</div> <div>translation;</div> <div>scale</div> </div>
Cost:	\$0.50
Manufacturing:	
Manufacturing Cost:	\$0.50
Operation:	
MTBF:	<model output>
Operation Task List:	-----
Operation Cost:	SUM(All Operation Time * Work Code RATE)
Maintenance:	
MTTM:	-----
Maintenance Task List:	-----
Maintenance Cost:	SUM(All Maintenance Time * Work Code RATE)
Repair:	
MTTR:	-----
Repair Task List:	-----
Repair Cost:	SUM(All Repair Time * Work Code RATE)

Figure 5 - Example Assembly List Frame

The frame shown in the figure has spaces for the following data:

- part number,
- part name,
- a reference to the frame's parent part,
- a list of component parts,

- coordinate transformations to translate each of the component's coordinate systems to the local coordinate system,
- a list of machining operations containing references to generic tasks with supplementary data such as special tool requirements and task durations,
- a list of assembly operations,
- a manufacturing cost model,
- a list of operations tasks,
- an operations cost model,
- a list of support tasks, and
- a support cost model.

Although the data shown in Figure 5 were considered important to the Phase I example, the frame structure created for a design will be dependent on the needs of the designer.

The designer of a system creates the frames for a system based on the analyses needed, through the selection of appropriate primitive data types. The example frame in Figure 5 consists of a numeric data field for the part number, a character field for the part name, a pointer to the parent component's frame, a list of pointers to the component frames, numeric fields for coordinate transformations, and a series of lists of pointers combined with numeric fields.

Frames contain either: 1) the coordinate transformations required for the union of the geometries of the component parts, 2) a reference to a CAD drawing file if there are no components, or 3) both if there are components and some type of interface data at that level. Drawings are created in the CAD program, the second major architectural component of the system. Macro commands to the CAD system are developed to create unions of drawings using the coordinate transformation information.

An additional construct of the system that will help designers to manage the volume and types of data is called a layer. A layer is a term borrowed from architectural CAD systems to signify a logical group of data. In the same way that all wiring-related data for the design of a building can be grouped into a wiring diagram, the proposed system uses layers to group related data.

Layers provide the designer with alternative ways to organize data. Through these alternative groupings, the designers can more easily manage the design data and gain insight into the merit of a design.

The hierarchical data discussed in the frame description above are an obvious layer. These types of data, when grouped across frames, describe a Work Breakdown Structure or Bill of Materials. Alternative views of these data, such as tree structures are created automatically by the system using the contents of the frames. Figure 6 shows two screens from the Phase I proof-of-concept computer program with a frame describing the switch coil and its position in the switch bill of materials. By creating precedence and successor relationships in the manufacturing, operation, and support sections of a design's frames, a PERT/CPM layer can be established for each of the life cycle phases of the system.

Although the discussion of frames and layers above describes the development of frames from which layers are derived, the proposed system has the flexibility to derive partial frames from layers. If the designer defines a hierarchical layer by drawing a graphical bill of materials as arcs and nodes, the system automatically creates a series of frames corresponding to the nodes with parent and component relationships determined from the layer drawing.

By making the connections between frames and layers automatic, the system reduces data entry requirements and provides the designer with the ability to easily check for errors in design definition. The automatic linkage also allows the designer to enter data in the most intuitive way. Designers can define the relationship between components first and then describe each component in detail, or vice versa. Rather than impose a design process, the system allows the designer to specify a design the way he wishes.

The third major part of the architecture is a tool to allow easy integration of analysis modes. This tool, called Glue, is currently under development by KDT for the USAF Space Systems Division. The tool allows users to specify the order in which models are to be run and then create the interfaces between the models.

After the feature database has been created as frames, the designer specifies the order in which the analysis models are to be run. The interfaces between models are created by specifying where the data comes from and what they are used for. Data sources are: frames, output files from analysis models, external databases, or user inputs. Data

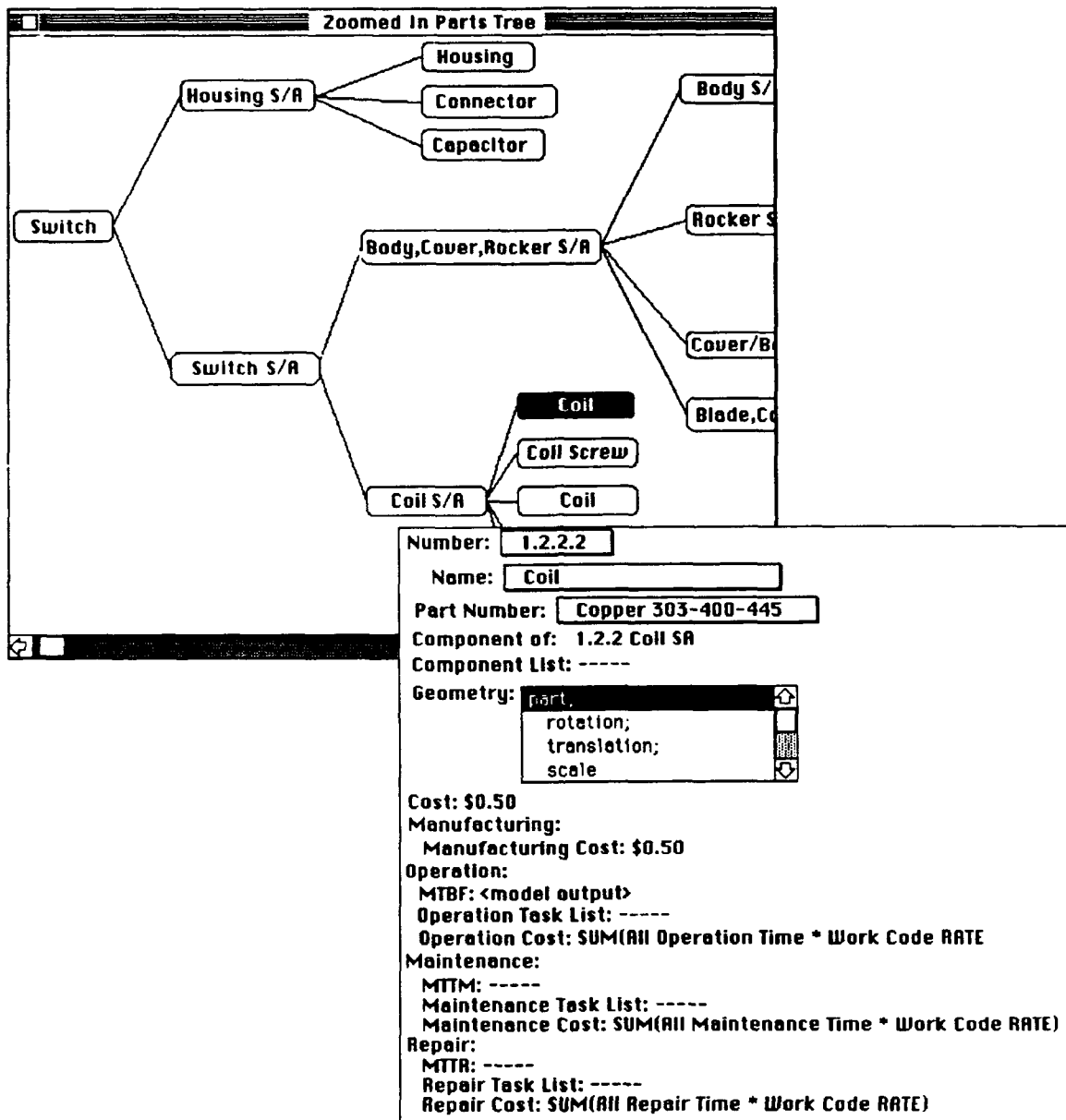


Figure 6 - Alternative Data Views

destinations are be: other analysis models, frames, or external databases. The run specification is developed in a workspace using nodes and arcs to define the run order for analysis models. Model interfaces are created by clicking on the arcs between models and defining the data sources and targets plus any transformations required to match data types.

The designer is then able to run the analysis and examine the results. The merit of the design is determined by an objective function or functions described by the designer, or influences outside the system.

Once the analysis has been specified, the system verifies that the model is complete. All inputs to models are checked to ensure that they either exist or will be created by the time they are needed.

After reviewing the results, the designer determines if the cost and performance of the design are adequate. If they are adequate, the initial design is complete. If the design is not adequate, the designer revises the features describing the design and reanalyzes the design in an iterative fashion until the design is adequate.

During the design revision process, the designer may wish to access the final major part of the system architecture, the advice system. The advice system contains a collection of rules about system behavior. The rules are derived from expert opinion and analysis of historical data. As previously stated, the scope of the project is not an attempt to develop an automatic learning system. The user interface of the expert system asks questions of the designer about the new knowledge to allow the basic rules to be updated.

When querying the advice system, the designer chooses advice from a number of topics. Rules in the system defined for Phase II have a tag describing the kind of information they concern. This tag will relate to a keyword list of topics maintained in the system. The designer is able to enter the network of rules and navigate around the network in order to obtain a feel for the best means to revise the design.

The program allows the designer to modify the hierarchy of the advice topics, adding new topics as required with no advice from the system on optimum organization of the keyword hierarchy. Search capabilities will be limited in Phase II to brute force search by the designer. Rather than specify a search by keyword sets and receive a list of rules to be

considered, the designer will consult the keyword topics he sees fit. The Phase II system will maintain a single monolithic rule set for each design.

Phase III activities will expand the capabilities of the advice system through improved search techniques and the segmentation of the rule set into topics about specific designs, with search across domain.

One feature of the expert system will be the development of a module to assist with experimental design for Taguchi type analyses. This module will ask the designer questions about the design to suggest a factorial of experiments. The results, when input by the designer, will be used as a part of the design analysis and provide suggestions for design changes.

As can be seen from the descriptions of the system, the Graphical User Interfaces (GUIs) that will be developed for the system will be critical to its success. The different views of the data, the ease with which frames can be designed, linked, and populated, the ease of defining analysis runs creating interfaces, and the ability of designers to modify and access the advice system will determine both the friendliness and power of the system. To a great degree the system will be a user interface, allowing designers to increase their productivity through easier access to data and analysis.

PROOF-OF-CONCEPT SYSTEM

A proof-of-concept computer program was developed as a part of Phase I of the project to show that the proposed system would prove useful to design engineers and that it is possible to develop such a system.

The program was developed in the C programming language under the Macintosh Programmer's Workbench (MPW) for the Apple Macintosh, and is limited primarily to definition of the user interface requirements. The POC program does not link to a database management system, CAD program, or expert system. The capabilities simulated for each of those systems in the POC program will be created in the programming languages of the respective systems.

The design example used in Phase I is a microwave switch manufactured by Arrowsmith Shelburne, Inc. (ASI), a subsidiary of KDT Industries, Inc. The switch allows microwave

signals to be switched between either of two poles and a common ground. The CAD drawings in the CAD layer of the POC program were developed in AutoSolid, printed, scanned and converted to Macintosh PICT file format. Solid models and geometric translations for all components of the switch were developed in IRIT, a public domain solid modeling system developed at the University of Utah. AutoSolid was used to create drawings because of limits in IRIT capabilities for combining component drawing files. Life cycle analysis models for the switch were created from basic physical properties of switch components and from interviews with ASI personnel. Expert system rules were also derived from interviews with ASI personnel.

PHASE II PLAN

The plan for Phase II development of the proposed system consists primarily of technical and management objectives and a workplan of tasks and schedule for achieving those tasks. The objectives and workplan are described in the Phase II Proposal for this project.

RESULTS

The major results of Phase I of the Project were:

- a methodology and architecture for a concurrent feature based modeling system,
- a limited proof-of-concept computer program,
- an example of a concurrent feature based design for a microwave switch, and
- a plan for extending the Phase I results to a commercial product.

The methodology and architecture developed as a part of Phase I are described in the Description of the Work section of this report. The plan for accomplishment of Phase II is the subject of the Phase II proposal. The proof-of-concept computer program and microwave switch example are described in detail in Appendix A, the User's Guide, and are summarized below.

PROOF-OF-CONCEPT PROGRAM

The limited proof-of-concept system was developed in the C programming language on the Apple Macintosh. The system and its operation are described in the User's Guide attached as Appendix A.

The proof-of-concept system has three major features:

- layers,
- performance/cost analysis, and
- advice.

Layers

The layers implemented in the POC system include: 1) bill of materials layers, 2) a CAD drawing layer, and 3) a frame layer created for a microwave switch. The bill of materials layer describes the component structure of the switch in graphical and textual form. The graphical representation of the bill of materials has been implemented for the POC in two layers (Figure 7). These layers are zoomed-in and zoomed-out views of the component structure of the switch. The POC system was created with two layers due to programming constraints, while the Phase II system will have the ability to zoom in and out of layers of

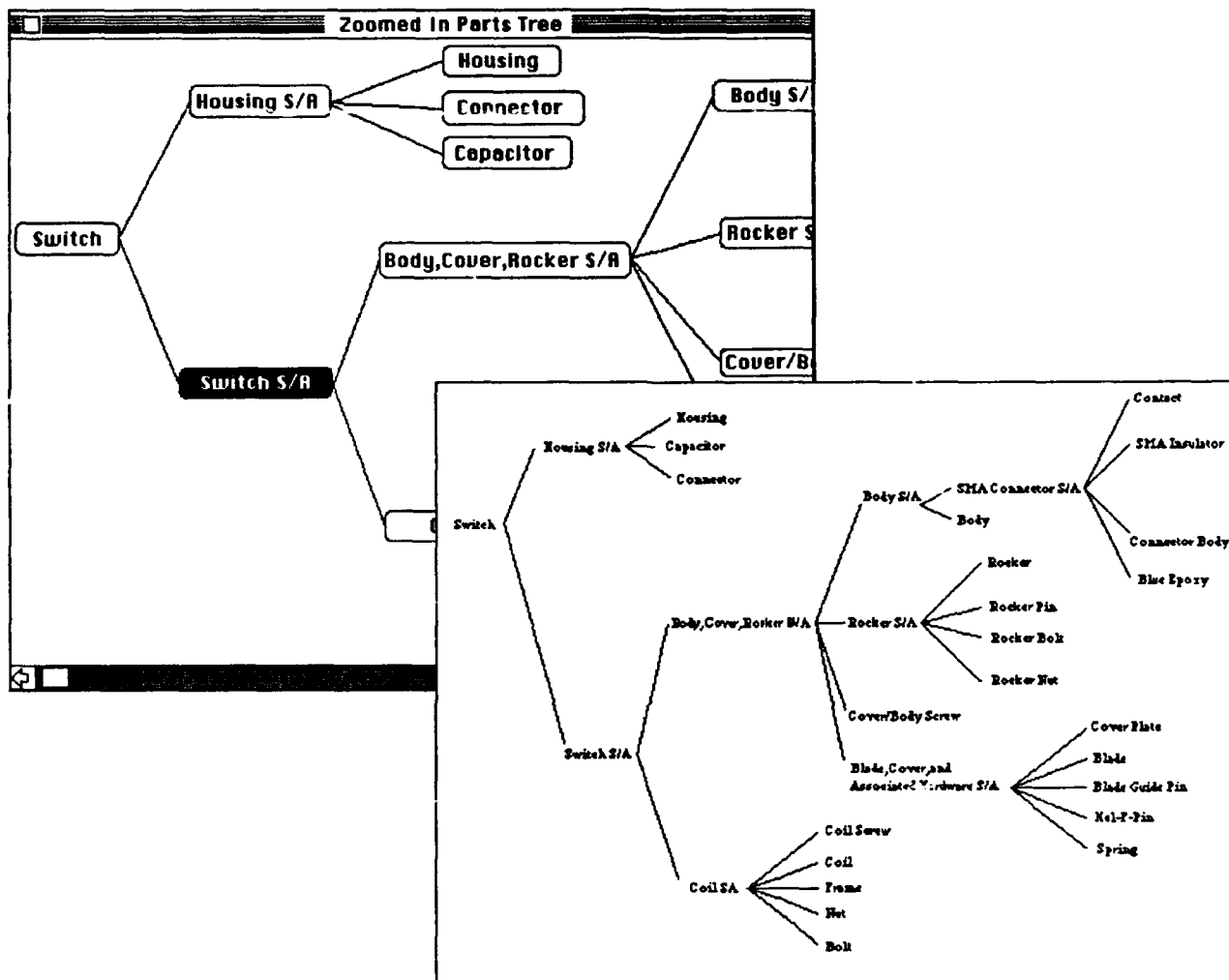


Figure 7 - Proof-of-Concept Bill of Materials

this type. Components are selected in this (zoomed-in layer) by clicking on the node containing the component name.

A textual version of the switch bill of materials constitutes another layer of the system. This layer shows the hierarchy of components by levels of indentation (Figure 8). All components at the same level in the parts hierarchy are indented to the same degree. Components are selected in this layer by clicking on the radio button to the left of the component name.

CAD drawings for selected components are grouped in a layer. These drawings were created from solid models in AutoSolid, but are not live CAD drawings for the POC system.

The final layer implemented in the POC system is the frame layer. This layer contains frame structures for the component parts. The frame layer references external databases for materials, parts lists, and task descriptions for several of the frames.

The layers of the POC system are connected in the sense that selecting a component in the bill of materials layer causes that same component to be selected in the other layers. If the blade component of the switch is selected in the graphical bill of materials, the blade CAD drawing will be presented if the CAD layer is selected, and the blade frame will be shown if the frame layer is selected.

Bill of Materials	
<input type="radio"/>	1.0 Switch
<input type="radio"/>	1.1 Housing S/A
<input type="radio"/>	1.1.1 Housing
<input type="radio"/>	1.1.2 Connector
<input type="radio"/>	1.1.3 Capacitor
<input checked="" type="radio"/>	1.2 Switch S/A
<input type="radio"/>	1.2.1 Body,Cover,Rocker S/A
<input type="radio"/>	1.2.1.1 Body S/A
<input type="radio"/>	1.2.1.1.1 SMA Connector S/A
<input type="radio"/>	1.2.1.1.1.1 Contact
<input type="radio"/>	1.2.1.1.1.2 SMA Insulator
<input type="radio"/>	1.2.1.1.1.3 Connector Body
<input type="radio"/>	Blue Epoxy
<input type="radio"/>	1.2.1.1.2 Body
<input type="radio"/>	1.2.1.2 Cover/Body Screw
<input type="radio"/>	1.2.1.3 Blade,Cover,Hardware S/A
<input type="radio"/>	1.2.1.3.1 Cover Plate
<input type="radio"/>	1.2.1.3.2 Blade
<input type="radio"/>	1.2.1.3.3 Blade Guide Pin
<input type="radio"/>	1.2.1.3.4 KEL-F-Pin
<input type="radio"/>	1.2.1.3.5 Spring
	<div>Page</div>

Figure 8 - Example Bill of Materials Layer

Analysis

The proof-of-concept system has several hard coded analysis models of switch performance and cost. The order in which the models are solved is shown in Figure 9.

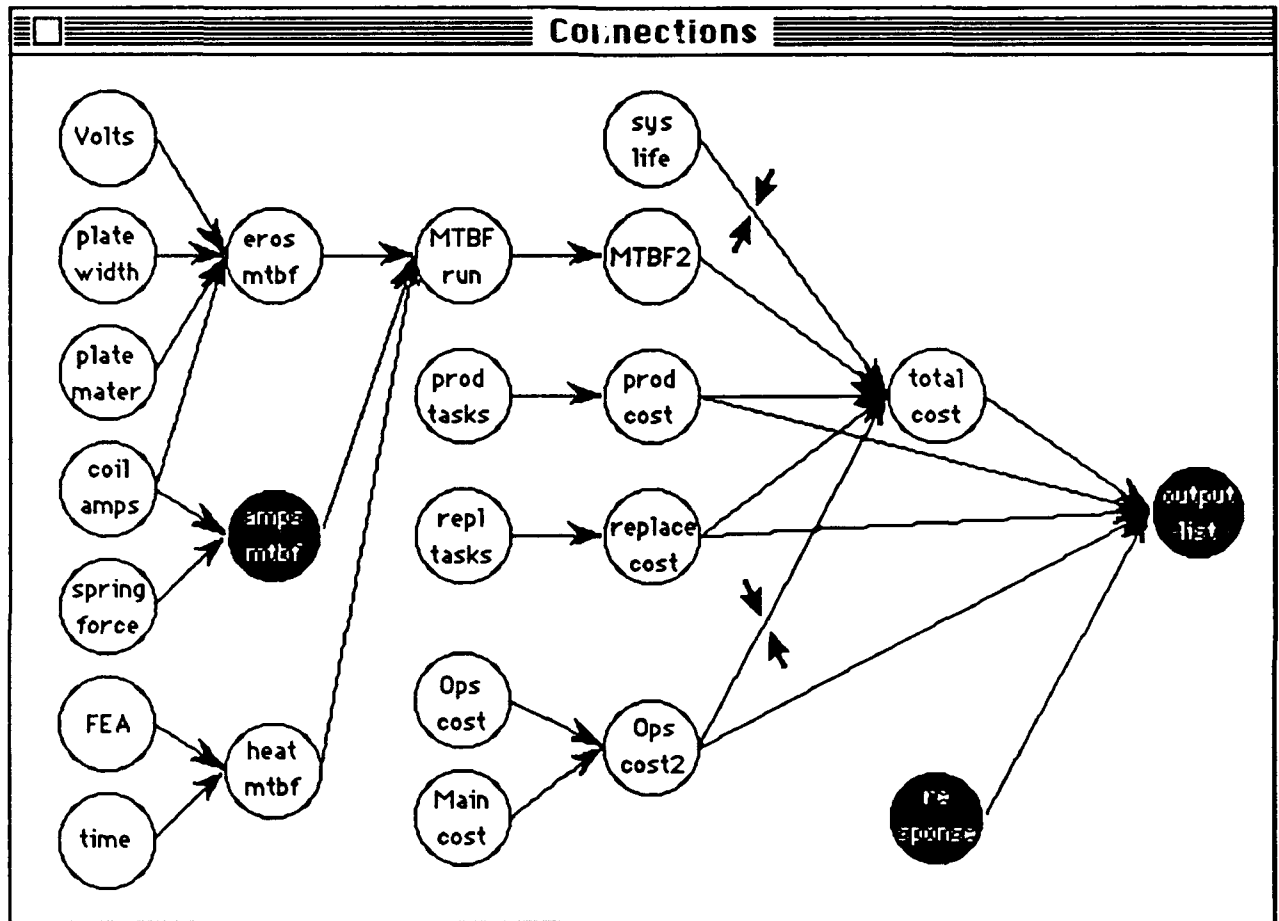


Figure 9 - Model Analysis Order

This figure demonstrates the look of the analysis specification section of the proposed system. Models will be specified as nodes, with arcs indicating: 1) the order of analysis of the models, and 2) the structure of the data interfaces between models.

When the POC user selects the RUN option, the models are evaluated using the current design of the switch as defined in the frame layer. The analyses use materials thicknesses, part properties, and costs for the current features of the design. The analysis results are then written to a log giving a historical summary of results.

Advice

Advice is implemented in the POC for a subset of switch features. Advice for switch design can be obtained by selecting the ADVISE option. The user is presented with a list of topics for which advice exists (Figure 10). Selecting a topic gives a list of subtopics for which there is advice. Advice for the POC is limited to information about how a change in a feature will affect cost and performance. The user can move up and down in the hierarchy of advice, obtaining information about how changes in features will change performance and cost. Changing values of features under the Advice option does not change the actual feature value. The designer must exit the Advice section of the POC and change the value of the feature in the frame layer.

MICROWAVE SWITCH APPLICATION

The microwave switch example developed for Phase I illustrates the potential complexity of even a limited model. The example had a hierarchy of nearly 50 components, four models of Mean Time Between Failure (MTBF) performance, a switch response time performance model, and cost models for manufacturing, operation, maintenance, and replacement.

The switch has two poles and a common ground. Switching is accomplished by applying current to one of two electromagnetic coils. The energized coil attracts one end of a metal rocker bar depressing the other end. The depressed end of the rocker presses a pin moving a blade that connects one of the poles and the common ground.

The description of the switch is important to the project primarily in the types of models and data that must be accommodated, data representations needed to easily manipulate design features, the requirements to develop features, and the advice needs for a designer. The data types discussed in the Description of the Work section of this report were sufficient to represent all of the features that were determined to be important for the POC. The only major additions to standard data base types were lists, pointers, and coordinate transformations.

In addition to the data types identified in the POC, layer types were created for the switch example. Nodes and arcs were useful for representing hierarchical and precedence relationships as well as influences between features. Radio buttons, check boxes, and

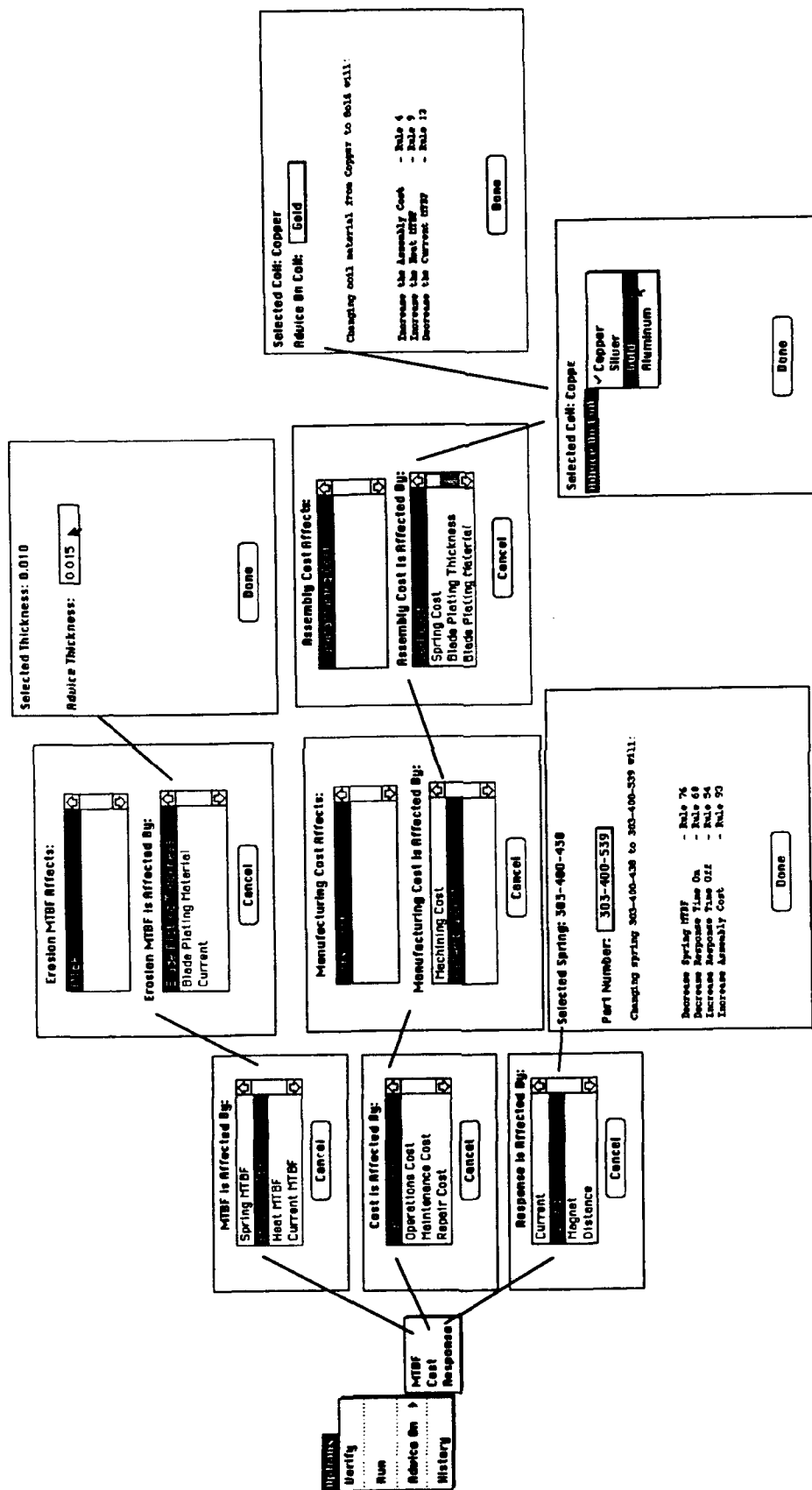


Figure 10 - Advice from Proof-of-Concept Program

scrollable lists were found useful for creating frames and text representations of node and arc diagrams.

Several design heuristics were developed for the POC from interviews of design experts and as a result of the analysis model development process. A simple advice network in which the designer enters a list of design rules and asks questions about how design changes will effect performance, was found to provide surprising amounts of design assistance. The relative simplicity of the assistance system for modification and access of advice, and the amount of information available to the designer, have driven the design of the advice system.

POTENTIAL APPLICATIONS

The proof-of-concept example shows that it is possible to determine the data types and representations needed to model an electromechanical part. The menus and program structure demonstrate that the tool can be useful for design engineers. The Phase I plan shows how the functioning system can be developed.

The application proposed for Phase II is to develop a feature based life cycle model of the wiring system for an improved F-16 bomb rack, using solid state componentry. The approach is of interest because the design promises better performance, better reliability, lower maintenance cost, and higher manufacturing costs. Because of its higher initial cost, procurement of an improved bomb rack would be contingent on showing better life cycle cost effectiveness.

Beyond the F-16 bomb rack there is the potential to improve the design of a wide range of mechanical parts. The generality of the system will not limit it to a specific part type. The limits on its effectiveness are only in the knowledge stored in the expert assistance part of the system. Expert opinions can be entered by the designer from his own experience or obtained through interviews with experts as they were for the POC example.

This tool can play a significant role in the communication of ideas between the design, manufacturing, and operations communities. By allowing an industrial engineer to see the product from the perspective of the manufacturing engineer and the logistician, the tool will create an improved sense of the importance of each team member's contribution to the life cycle value of the product. As this systemic perspective is achieved, each member of the team will have a stake in maintaining an accurate representation of the product life cycle data in the tool at any given time. The tool is meant to be a catalyst for the learning process that characterizes good design methodology. As lessons are learned, the tool is designed to capture the new data and relationships between the data. This capability has application in any industry in which products must be designed for life cycle performance. The groups to be targeted for the initial user community for this tool in the private sector include the developers of satellite components, medical equipment, military hardware, transportation systems, and communications equipment.

Bibliography

Shoemaker, A. C., Kacker, R. N.: "A Methodology for Planning Experiments in Robust Product and Process Design," published in *Quality and Reliability Engineering International* by John Wiley & Sons, Vol. 4, No. 2, April - June 1988.

Box, George, Bisgaard, Soren, and Fung, Conrad: "An Explanation and Critique of Taguchi's Contributions to Quality Engineering," published in *Quality and Reliability Engineering International* by John Wiley & Sons, Vol 4, No. 2, April - June 1988.

Winner, R. I., Fennell, J. P., Bertrand, H. E., and Slusarczyk, M.M.: "The Role of Concurrent Engineering in Weapons System Acquisition," DA report R-338 submitted to OASD P&L, October 1988.

Bailey, M.W., and Overton, K.S., "Automated Aircraft Engine Costing Using Artificial Intelligence," presented at the 26th Joint Propulsion Conference, paper AIAA 90-1887, July 1990.

Juarez, J.M., Gunther, Eric J., and Toomey, John E.: "Quality function Deployment Applied to an ALS Cryogenic Tank," presented at the AIAA Aerospace Engineering Conference & Show, paper AIAA-90-1807, February 1990.

Ryan, N. E. (Editor): "Taguchi Methods and QFD: Hows and Whys for Management: a special collection of papers and essays on today's quality issues and new quality technologies", ASI Press, 1988.

Juricic, D. and Barr, R. E.: "New Methodology for Developing and Conveying Design Ideas," proc. of the Meeting of the Engineering Design Graphics Division of the ASEE, November 1989.

Barr, R. E., and Juricic, D.: "Some Geometric Operations for Solid Modeling," *The Engineering Graphics Journal*, Vol 53, No.3, Autumn 1989.

Llewelyn, A. I.: "Review of CAD/CAM, The CAD/CAM Centre, pp 1 - 5.

Lawlor-Wright, T. F., and Hannam, R. G.: "A feature-based design for manufacture: CAD/CAM package," (Lawlor-Wright - ICL, ITL, England and Hannam - UMIST, Manchester, M60 1QD, England), 1989.

Mills, R.: "Today's FEA: Integrated, Accurate, and Fast", published in *Computer-Aided Engineering*, January, 1990.

Shah, J. J.: "Feature transformations between application - specific feature spaces," published in *Computer-Aided Engineering Journal*, Dept of Mechanical & Aerospace Engineering, Arizona State University, Tempe, AZ, 1988.

Pande, S.S., and Palsule, N.H.: "GCAPPS - a computer-assisted generative process planning system for turned components," Department of Mechanical Engineering, Indian Institute of Technology, Bombay, India, 1988.

"Computer-Aided Systems and Software Engineering Products for Time-Critical Applications Development", Report prepared for NADC by Software Productivity Solutions, April 1988.

Sievert, G.E., and Carrio, Jr, M.A.: "The Case for Software Engineering Environment," 1986.

"Pro/ENGINEER The Parametric, Feature-based Solid Modeling System," Parametric Technology Corporation, 128 Technology Drive, Waltham, MA, 1989.

Hardwick, M., Spooner, D., Hvannberg, E., Downie, B., et al.: " ROSE: A Database System for Concurrent Engineering Applications," Rensselaer Design Research Center & Dept of Computer Science Rensselaer Polytechnic Institute, Troy, NY, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Lewis, J.W.: "An approach to applications Integration for concurrent Engineering," GE Research & Development, Schenectady, NY, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Brown, D. R., Cutkosky, M. R., and Tenenbaum J. M.: "Next-Cut: A Computational Framework for Concurrent Engineering," (Brown, Mechanical Engr Dept - University of Utah, Cutkosky, Center for Design Research, Stanford University, Tenenbaum, Schlumberger Technologies Corporation & Computer Science Department, Stanford University), Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

LeClair, S.R.: "A Coordinated, Feature-Based Rapid Design System," Manufacturing Research, Air Force Materials Laboratory, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Safier, S.A., and Fox, M.S.: "The Role of Architecture in Computer-Assisted Design Systems," Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Kannan, R., Cleetus, K.J., and Reddy, Y.V.: "The Local Concurrency Manager in Distributed Computing," Concurrent Engineering Research Center, West Virginia University, Morgantown, WV, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Lohr, P.J., Czechowski III, J., Lewis, J.W., and Chiang, Wei-Shan: "An Application Integration Wrapper for DICE," GE Corporate Research and Development, Schenectady, NY, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Mistree, F. and Muster, D.: "Conceptual Models for Decision-Based Concurrent Engineering Design for The Life Cycle," Professor, Dept. of Mechanical Engineering, University of Houston, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Phillips, R.E., and Aase, J.: "An Integrated Environment for Concurrent Engineering," Advanced Computer Aided Engineering, GE Aircraft Engines, Lynn, MA 01910, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

De Fazio, T. L., Edsall, A. C., Gustavson, R. E., et al., "A Prototype for Feature-based Design for Assembly," The Charles Stark Draper Laboratory, Inc, Cambridge, MA 02139, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Elliott M. W.: "Reliability Analysis Expert System in Concurrent Engineering," Automation Sciences Laboratory, Northrup Corporation, Palos Verdes Peninsula, CA 90274, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

Hill, R.R.: "Enhancing Concurrent Engineering Using Quality Function Deployment Based Tools," Air Force Human Resources Laboratory, Wright-Patterson AFB, OH, Proc. of the Second National Symposium on Concurrent Engineering, February, 1990.

PDES, Inc., Presentation, August 1988.

Appendix A

Software User's Documentation:

ULCE Unified Life Cycle Engineering



Phase I SBIR Proof-of-Concept

August 1990

K_DT

Industries, Inc.

PREFACE

The Software User's Documentation for the ULCE Phase I Proof-of-Concept was prepared by the Advanced Systems Group of KDT Industries, Inc.

Questions concerning the technical content of this document, or inquiries in general about Unified Life Cycle Engineering, should be directed to:

Patrick R. Flanagan
KDT Industries, Inc.
Advanced Systems Group
P.O. Box 1787
Austin, Texas 78767

(512) 474-6312
(512) 473-8534 *fax*

This document and the accompanying Phase I Proof-of-Concept software package were prepared under Air Force SBIR contract number F33615-89-C5733.

CONTENTS

	Page
Introduction.....	1
Getting Started	1
System Requirements	1
Switch Background.....	2
Disclaimer.....	3
ULCE	4
File Menu	5
Edit Menu.....	7
Options Menu	8
Verify.....	8
Run.....	8
Advice On.....	9
MTBF.....	9
Cost.....	11
Response.....	14
History	16
Align Menu	17
Font Menu	18
Layer Menu.....	19
Parts Tree.....	19
Zoomed Out	19
Zoomed In	19
Bill of Materials	22
CAD Drawing	22
Assembly List	22
Connections	27
FEA.....	28
Toolkits	29
Sample Session	30
Appendix A.....	A-1
Quick Reference Card	

Introduction

KDT Industries, Inc. Unified Life Cycle Engineering (ULCE) design system effort is a research and development project aimed at integrating Feature Based Modeling (FBM) with Life Cycle Analysis (LCA).

Computer Aided Design (CAD) is evolving beyond its traditional function of automated drafting to Feature Based Modeling. FBM identifies features or attributes of a design for use in analytical models of system performance. These features may include the geometrical aspects of a design that are traditionally identified by CAD as well information such as material properties, costs, assembly tasks, surface attributes, operating procedures, and support requirements.

Life Cycle Analysis is an integral part of concurrent or simultaneous engineering. LCA examines important factors from all phases of the life of a system during its design. Examining these factors early in system life is an attempt to improve the cost effectiveness of the system. Cost effectiveness measures both the system cost and its ability to perform its intended function. Modeling the cost and effectiveness of a system over its lifetime is a very complex undertaking both in terms of the types and volumes of data required. This complexity motivates the need for a computerized system to store, analyze, and display information.

The ULCE development effort centers around determining the type of information needed to model system life cycles, storing the data efficiently, allowing the designer to analyze the data in ways that are meaningful to him/her, and presenting the data in a clear, unambiguous fashion.

Getting Started

ULCE is a regular Macintosh application and can be started as any other application by a double click on the ULCE icon:



ULCE uses standard Macintosh "look and feel" features such as mouse input and pull down menus. As with most Macintosh programs, ULCE is fairly easy to use without much formal explanation (i.e. "plug and play"). However, to fully understand the complexity and power of the ULCE system, one should at least review the enclosed quick reference card before attempting to use ULCE.

System Requirements

The Phase I proof-of-concept software package requires a Macintosh computer with at least 512K internal memory, and an 800k floppy disk drive.



Execution on a Macintosh II is strongly recommended as some screen formats were designed to use the larger screen.

The distribution disk contains the ULCE application file, this user documentation in a Microsoft Word data file, and several saved examples. ULCE can be executed directly from the distribution disk, or may be copied to the hard disk and executed from there.



While the program can be run from the floppy disk drive, it is recommended that the ULCE program be executed from the hard drive to increase execution speed.

Switch Background

This Phase I demonstration of the ULCE methodology uses an RF switch as the system being designed. The switch was chosen for many reasons, but there were three (3) primary reasons:

- 1) *It has an important function within the U.S. Air Force.* RF switches are used on Air Force Electronic Counter Measures (ECM) equipment and aircraft test sets. In addition to their importance in supporting the Air Force mission, the critical nature of ECM has generated Air Force interest in developing methods by which the Mean Time Between Failures (MTBF) for these switches can be extended.
- 2) *It demonstrates many of the important features of Unified Life Cycle Engineering.* The RF switch is a complex assortment of parts with varying levels of manufacturing, assembly, and operations cost. It is composed of both mechanical and electrical parts, allowing modeling of operation, failure, and cost. Although a single switch is small, it is relatively complex, containing approximately 50 component parts. This complexity makes this Proof-of-Concept more than just a demonstration of a "toy" problem. The switch used for the Proof-of-Concept is also comparatively inexpensive. This may allow alternative designs developed from a Phase II model to be produced and tested to further demonstrate the value of the ULCE system.
- 3) *It is an "in-house" product which ensures access to accurate design information.* The switch was also chosen to demonstrate the concept because it is produced by another subsidiary of KDT Industries, Inc. This relationship has allowed relatively free access to switch design drawings and other product information.

For this Phase I demonstration, three features of the switch are modeled: 1) the cost, 2) the Mean Time Between Failure (MTBF), and 3) the response as a function of time. For these three models, the user has access to and can change:

- 1) the current applied to the coil of the switch,
- 2) the material the coil is made of,
- 3) the material the blade is plated with,
- 4) the thickness of the plating material, and
- 5) the mass and spring constant of the spring.

Disclaimer

This manual makes the assumption that the user is familiar with the execution of at least one Macintosh application, and knows how to use the Macintosh mouse. The user should also have a working knowledge of how to select menus and be familiar with the different types of windows found in Macintosh applications. A Macintosh owner's manual is the best reference for further information on the Macintosh system and environment.

The ULCE software is delivered on an "as is" basis. While executing the program, one should remember that this is a limited demonstration of a very complex system. Every attempt has been made, through extensive testing and modification of memory requirement algorithms, to make the execution of the ULCE program as error free as possible. However, due to the nature of the Macintosh environment, there is a small chance that the application will stop, or "bomb", unexpectedly. If an unexpected interruption does occur, simply restart the ULCE program, as these random interruptions have no permanent effects on either the ULCE software, or the Macintosh environment.



It is highly recommended that all non-ULCE windows be closed prior to executing ULCE to free memory, and it is more aesthetically pleasing to run ULCE without background window updates while under MultiFinder

ULCE

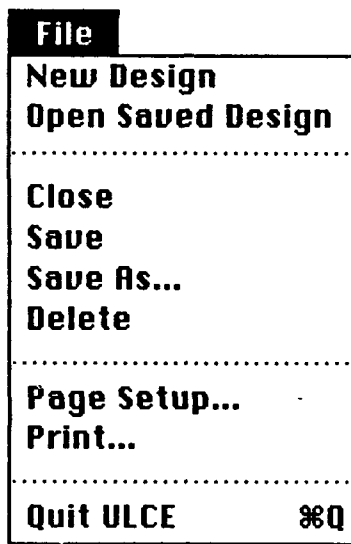
Upon entering the ULCE program, the "About ULCE" window appears. To proceed, click once anywhere within this window. The user is now at the top level of the ULCE program, with the following menu bar present:



Each of these menu items is discussed briefly below.

- Apple -** This is the standard Macintosh apple menu which allows access to the finder information and the desk accessories.
- File -** This is a standard Macintosh menu which controls the opening and closing of the application and associated data files.
- Edit -** This is a standard Macintosh menu which controls the formatting and manipulation of text and blocks within the program and provides access to data from other programs.
- Options -** This is an ULCE specific menu which allows access to the main functions within the ULCE program.
- Align -** This is an ULCE specific menu which is used to align text and fields as they are created.
- Font -** This is a standard Macintosh menu which is used to specify the desired character type and size for text and fields.
- Layer -** This is an ULCE specific menu which is used to display a component of the design, or the entire design structure, in any one of several different layers or views.

Options under each of these menu items for executing the ULCE program are presented in detail on the following pages.



New Design -

Not implemented for Phase I. In Phase II, this option will create a new ULCE session. The New command will be dimmed if a session is already open as only one session can be opened at a time.

Open Saved Design -

Opens other ULCE files stored on the floppy or the hard drive. Selecting Open will cause the standard Macintosh open display to pop-up with a list of the available ULCE files which may be loaded. It will also show which disk and directory are currently being scanned, along with the available storage memory.



Only ULCE type files can be opened from within the ULCE program, and ULCE files can not be opened by other applications.

Close -

Closes any and all active ULCE windows on the screen, but does not close the ULCE application after the final window is cleared.

Save -

Saves the current state of the ULCE program. If the current session is unnamed, a Save As dialog box appears so a save name for the file can be entered. For this prototype version of ULCE, each file saved requires 180 bytes of storage.

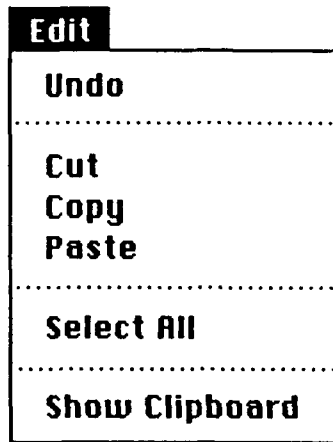
Save As -

Not implemented for Phase I. In Phase II, this option will allow a saved ULCE file to be saved under a different name.

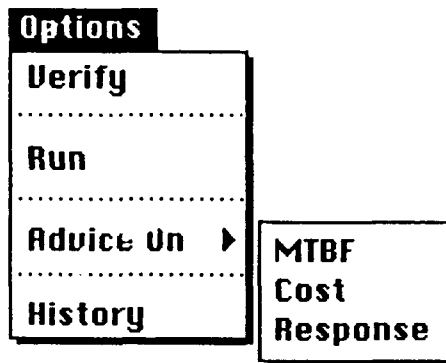
- Delete -** *Not implemented for Phase I. In Phase II, this option will remove the specified ULCE design file.*
- Page Setup -** *Not implemented for Phase I. In Phase II, this option will allow the user to set the page size, orientation, and other printing options.*
- Print -** *Not implemented for Phase I. In Phase II, this option will allow user specified layers or resource data to be sent to the printer.*
- Quit ULCE -** Used to exit the ULCE program and return to the finder. Quit can also be invoked at any time by depressing the command key ⌘ and the Q key at the same time.



Note that for this Phase I prototype, no check is made to see if changes should be saved prior to exiting the ULCE program.



- Undo -** *Not implemented for Phase I. In Phase II, this option will reverse the most recent action. Not all actions will have the undo option.*
- Cut -** *Not implemented for Phase I. In Phase II, this option will remove the active selection and place it on the clipboard.*
- Copy -** *Not implemented for Phase I. In Phase II, this option will duplicate the active selection and place it on the clipboard. Copy will not remove the selection from the active window.*
- Paste -** *Not implemented for Phase I. In Phase II, this option will copy the contents of the clipboard into the active window at the insertion point specified.*
- Select All -** *Not implemented for Phase I. In Phase II, this option will activate all items in the current window.*
- Show Clipboard -** *Not implemented for Phase I. In Phase II, this option will display the current contents of the clipboard.*

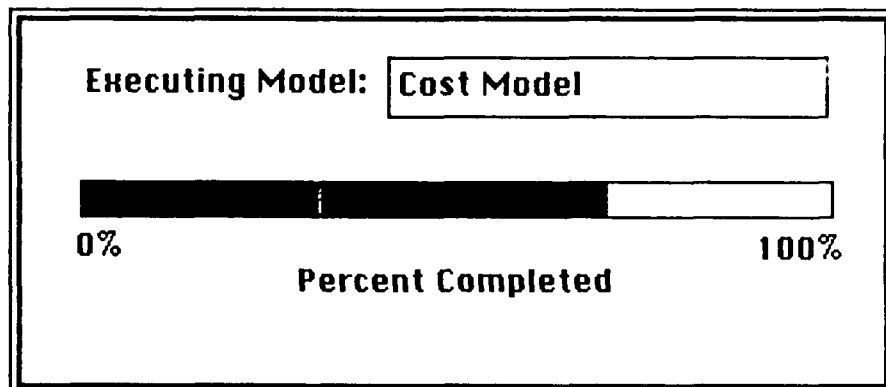


Verify -

Not implemented for Phase I. In Phase II, the verify option will check the syntax of the commands which invoke the user defined models and will see that each input file required for model execution has an established data link to it.

Run -

Used to invoke the user specified models with the user defined input data. When chosen, all ULCE windows are closed and the run status screen appears displaying the current model being executed:



For the Phase I proof-of-concept, there are three (3) active models for the switch example:

- 1) Cost of the System
- 2) Mean Time Between Failure
- 3) Response Time



Note that for this Phase I proof-of-concept, once a run is started there is no way to stop it until it has completed all defined models!

Upon completion of the run, the performance output screen appears:



Performance Output	
Total Cost of System: \$2922.33	
Production Cost: \$49.96	
Ops and Maintenance Cost: \$106.69	
Replacement Cost \$1.75	
Response of System:	
	On: 0.00349 (sec)
	Off: 0.02946 (sec)
MTBF: 146.3 hours	
Maintenance Cost: \$163.42	

Advice On -

Invokes a hierarchical menu of items which the user can ask the ULCE expert system questions about. For this Phase I proof-of-concept, the advice is limited to the affect that given features have on three performance characteristics modeled.

MTBF -

Provides information on the features which have an affect on Mean Time Between Failures. The first screen displays the different MTBF models for which advice is available:

MTBF is Affected By:	
Spring MTBF	 
Erosion MTBF	
Heat MTBF	
Current MTBF	
<div>Cancel</div>	

For this Phase I proof-of-concept system, only the Erosion MTBF advice is connected. A double click on the Erosion MTBF brings up the Erosion MTBF affects window:

Erosion MTBF Affects:

MTBF

Erosion MTBF is Affected By:

Blade Plating Thickness
Blade Plating Material
Current

Cancel

A double click on MTBF will return program control to the previous window. Clicking in the Cancel button in any window will cause the advice function to stop and program control to return to the top level. For this Phase I proof-of-concept program, the Blade Plating Thickness and Current are connected for advice. A double click on Blade Plating Thickness brings up the thickness advice screen:

Selected Thickness: 0.010

Advice Thickness: 0.015

Done

To obtain advice on the thickness, move the pointer into the "Advice Thickness" text edit box and click on the default value. Change this value to the desired thickness for advice and press <Return>. The advice for the thickness entered as compared to current thickness from the ULCE session appears as follows:

Selected Thickness: 0.010

Advice Thickness:

Changing thickness from 0.010 to 0.015 will:

Increase Blade Plating Cost	- Rule 41
Increase Erosion MTBF	- Rule 25

Done

Advice on another thickness may be obtained in the same manner. When no further advice on thickness is desired, clicking in the Done button will stop the advice function and return control to the top level.



The advice system for Current is invoked in the same manner as that for Thickness: change the Advice Current to the desired value and enter <Return>.

Cost - Provides information on the features which have an affect on the different costs of the defined system. The first screen displays the different Cost models for which advice is available:

Cost is Affected By:

Manufacturing Cost	↑
Operations Cost	
Maintenance Cost	
Repair Cost	↓

Cancel

For this Phase I proof-of-concept program, only the Manufacturing Cost advice is connected. A double click on Manufacturing Cost brings up the Manufacturing Cost affects window:

Manufacturing Cost Affects:

Total Cost	↑
	↓

Manufacturing Cost is Affected By:

Machining Cost	↑
Assembly Cost	
	↓

Cancel

A double click on Total Cost will return program control to the previous window. As with all of the advice screens, a single click in the Cancel button will cause the advice function to stop and program control will return to the top level. For this Phase I proof-of-concept prototype, only the Assembly Cost has been connected for advice. A double click on Assembly Cost brings up the Assembly cost affects screen:

Assembly Cost Affects:

Manufacturing Cost	↑
	↓

Assembly Cost Is Affected By:

Coil Cost	↑
Spring Cost	
Blade Plating Thickness	
Blade Plating Material	↓

Cancel

A double click on Manufacturing Cost will close the Assembly Cost effects window and re-open the Manufacturing Cost effects window. For Phase I, only the Coil Cost has been connected for advice. A double click on Coil Cost opens the Cost advice window:

Selected Coil: Copper

Advice On Coil:

✓ Copper
Silver
Gold
Aluminum

Done

Advice on different coil materials is obtained by clicking on the "Advice on Coil" pop-up menu and dragging the pointer (with the mouse button still down) to the desired material for which advice is wanted. Releasing the mouse will cause the expert system to display advice concerning the effect that changing the coil material will have on the defined system:

Selected Coil: Copper
Advice On Coil:

Changing coil material from Copper to Gold will:

Increase the Assembly Cost	- Rule 4
Increase the Heat MTBF	- Rule 9
Decrease the Current MTBF	- Rule 13

Advice on another coil material may be obtained by repeating the process of selecting a material from the "Advice on Coil" pop-up. When no further advice on the Coil material is desired, a single click in the Done button will stop the advice function and return control to the top level.

Response - Provides information on the features which have an affect on the response time of the defined switch system. The first screen displays the different response models for which advice is available:

Response is Affected By:

Current	↑
Spring	
Magnet	
Distance	↓

Cancel

For this Phase I proof-of-concept program, only the effect of the Spring on the response is connected to the advice system. A double click on Spring brings up the Spring response affects window:

Spring Response Affects:

Response	↑
	↓

Spring Response is Affected By:

Mass of Spring	↑
Spring Constant	
	↓

Cancel

A double click on Response will return program control to the response models window. For this Phase I proof-of-concept program, only the Spring Constant has been connected for advice. A double click on Spring Constant brings up the spring advice screen:

Selected Spring: 303-400-438

Part Number: 303-400-539

Changing spring 303-400-438 to 303-400-539 will:

Decrease Spring MTBF	- Rule 76
Decrease Response Time On	- Rule 60
Increase Response Time Off	- Rule 54
Increase Assembly Cost	- Rule 93

Done



The advice system for the Spring is invoked in the same manner as that for the Coil: click on the "Advice Spring" pop-up and drag to the desired spring and release the mouse.

History -

This option is used to view a comparison of the thickness, current, coil number, spring number, plating thickness, switch cost, MTBF, system cost, and average response time for the last 5 runs made with the current ULCE session:

History									
#	Thickness	Current	Coil	Spring	Plating	Cost	MTBF	Total	Avg Time
1	0.010	100	Cu	438	Al	49.96	157.7	2586.04	0.01648
2	0.010	100	Cu	438	Ag	50.10	120.5	4046.10	0.01674

In Phase II, this option will be connected to a more elaborate output mechanisms such as reports, charts, and graphs.

Align**Grids****Show Grids****Hide Grids****Snap To Grids****Grids -**

Not implemented for Phase I. In Phase II, this option will allow the user to define the size of the grid to which objects will be confined.

Show Grids -

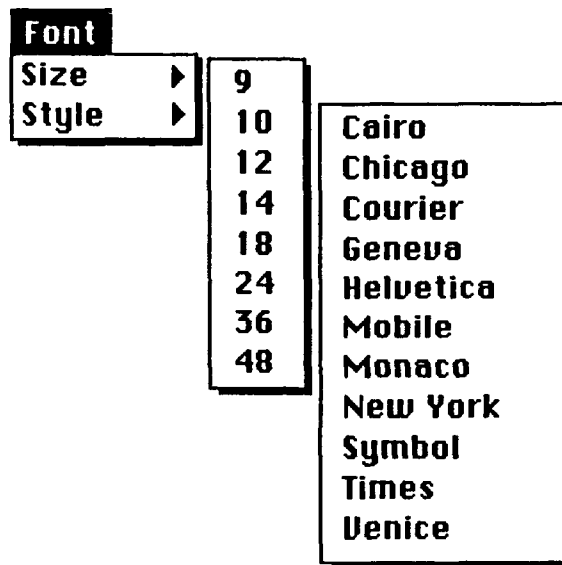
Not implemented for Phase I. In Phase II, this option will allow the user to display the defined grid system on the screen.

Hide Grids -

Not implemented for Phase I. In Phase II, this option will allow the user to remove the displayed grid system from the screen.

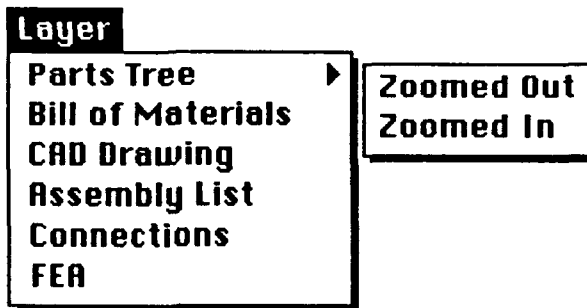
Snap To Grids -

Not implemented for Phase I. In Phase II, this option will allow the user to automatically align objects to the defined grid.



Size - Invokes a hierarchical menu of font sizes available. *The font sizes are not implemented for Phase I.*

Style - Invokes a hierarchical menu of font styles available. *The font styles are not implemented for Phase I.*



Parts Tree-

Invokes a hierarchical menu of views for displaying a complete breakdown of all of the parts associated with the RF switch design in a "tree" format. In phase II, this option will have full "zoom" capabilities at the control of the user. For this Phase I demonstration, only the fully "zoomed out" view and the fully "zoomed in" view are incorporated.

Zoomed Out - This option allows the user to see the entire tree breakdown of all of the parts associated with the design of an RF switch on a single screen. An example of the zoomed out parts tree layer is seen in Figure 1. There are no other views, or pop-up windows within this view. For Phase II, clicking the mouse pointer on a specific feature will "zoom in" to a close up view of that feature. This function has not been activated for this Phase I proof-of-concept program.

Zoomed In - This option greatly enlarges the parts tree diagram. The user can use this view to select the component of the switch design which the user desires to be the active component. (The *active* component is defined as the component which the CAD Drawing layer, and the Assembly List layer will automatically bring up specific information about when they are selected.) The current active component is highlighted in this view. An example of the zoomed in parts tree with the Switch S/A selected as the active component is seen in Figure 2. For this proof-of-concept model, the components which may be selected as the active design component are the:

- 1) Switch
- 2) Switch S/A
- 3) Body, Cover, Rocker S/A
- 4) Blade, Cover, and Associated Hardware S/A
- 5) Blade
- 6) Spring
- 7) Coil S/A
- 8) Coil

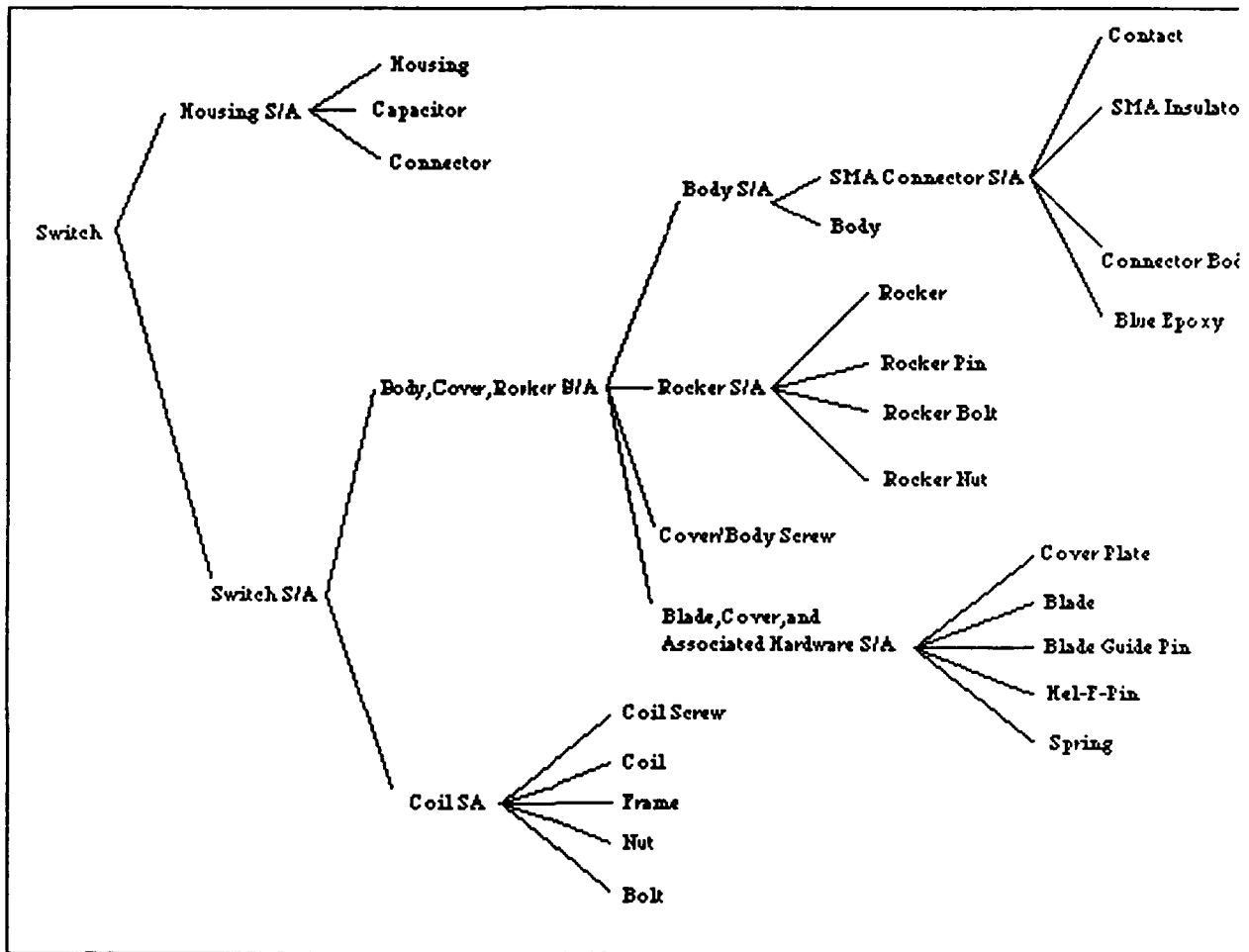


Figure 1 - Example Zoomed Out Parts Tree Layer

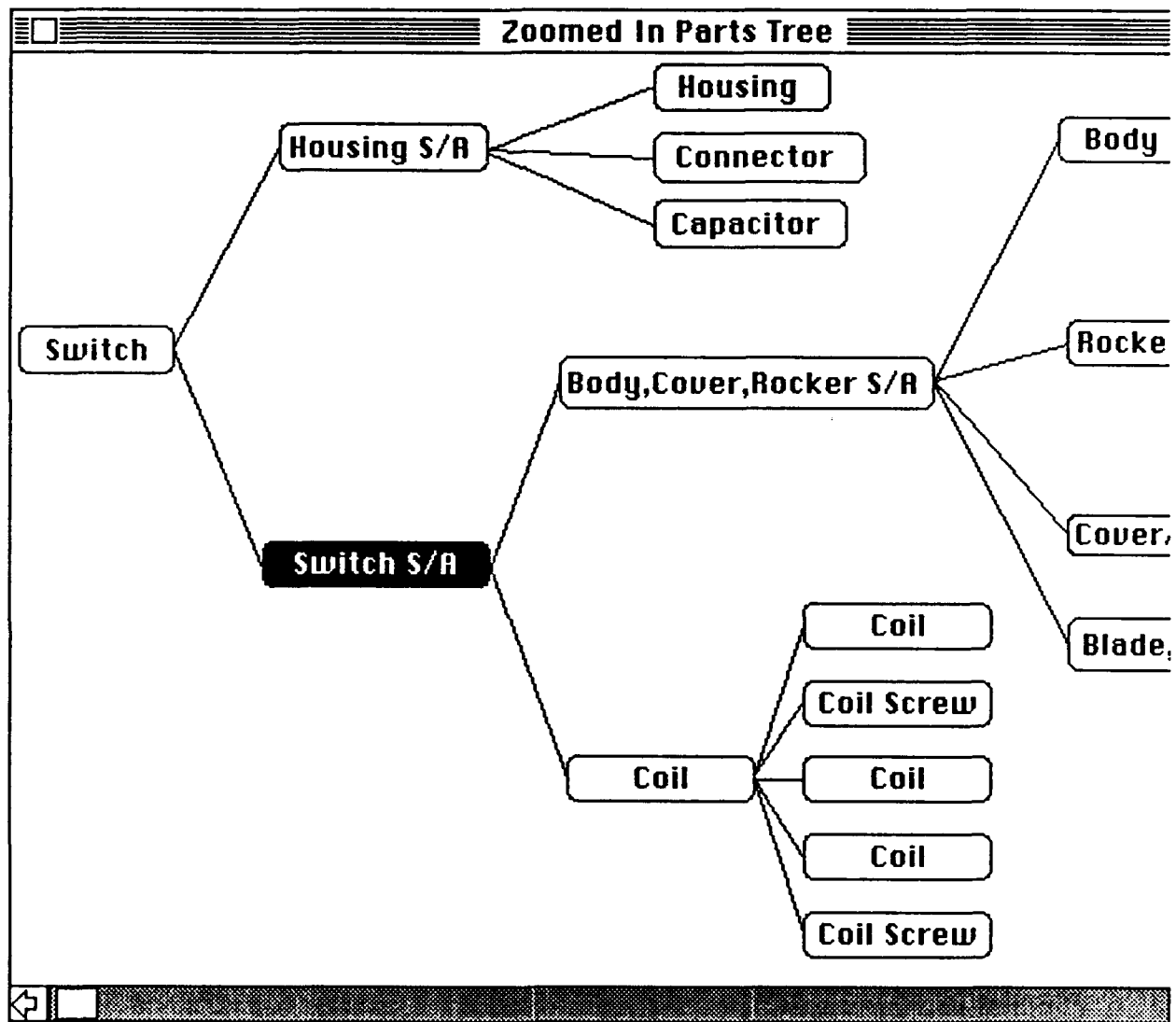


Figure 2 - Example Zoomed In Parts Tree Layer

Bill of Materials	
<input type="radio"/>	1.0 Switch
<input type="radio"/>	1.1 Housing S/A
<input type="radio"/>	1.1.1 Housing
<input type="radio"/>	1.1.2 Connector
<input type="radio"/>	1.1.3 Capacitor
<input checked="" type="radio"/>	1.2 Switch S/A
<input type="radio"/>	1.2.1 Body,Cover,Rocker S/A
<input type="radio"/>	1.2.1.1 Bcdy S/A
<input type="radio"/>	1.2.1.1.1 SMA Connector S/A
<input type="radio"/>	1.2.1.1.1.1 Contact
<input type="radio"/>	1.2.1.1.1.2 SMA Insulator
<input type="radio"/>	1.2.1.1.1.3 Connector Body
<input type="radio"/>	Blue Epoxy
<input type="radio"/>	1.2.1.1.2 Body
<input type="radio"/>	1.2.1.2 Cover/Body Screw
<input type="radio"/>	1.2.1.3 Blade,Cover,Hardware S/A
<input type="radio"/>	1.2.1.3.1 Cover Plate
<input type="radio"/>	1.2.1.3.2 Blade
<input type="radio"/>	1.2.1.3.3 Blade Guide Pin
<input type="radio"/>	1.2.1.3.4 KEL-F-Pin
<input type="radio"/>	1.2.1.3.5 Spring



Page

Figure 3 - Example Bill of Materials Layer

Bill of Materials -

This option allows the user to view the switch design session via an outline of the component parts. As with the zoomed in parts tree option, the active component of the switch design may be set and changed in this layer. The current active component will have a black dot to its immediate left. An example bill of materials layer with the Switch S/A as the active component is seen in Figure 3. Note that there are two (2) pages to the bill of materials screen. The pages are switched by clicking the mouse in either the up or the down page arrow. For Phase II, all of the Bill of Materials information will be on a single, scrollable screen.

CAD Drawing -

This option displays a CAD drawing of the current active component (set by the Bill of Materials option or the Zoomed In Parts Tree option) of the switch design. For this Phase I program, nothing more than displaying the picture is done by this option. The user is left to experiment with this layer and explore the different drawings of the switch design.

Assembly List -

This option is used to define the individual details of cost, manufacturing, assembly, operation, maintenance, and repair associated with a given component of the switch design. In addition, the components list of the component, and the geometry of the object are also defined in this layer. An example assembly list for the Coil is seen in figure 4.

Each assembly list displays the component diagram number (as specified in the Bill of Materials layer) and name at the top of the list. Clicking the mouse on the number will pop-up a menu of all of the diagram numbers. Selecting a number other than the current diagram number will cause the ULCE program to select that component diagram as the active component, and pass program control to the zoomed in view. This is the way that the part name pop-up menu functions also.

Number:	Name:
1.0	Switch
1.2	Switch S/A
1.2.1	Body,Cover,Rocker S/A
1.2.1.3	Blade,Cover,HW S/A
1.2.1.3.2	Blade
1.2.1.3.5	Spring
1.2.2	Coil S/A
✓ 1.2.2.2	✓ Coil

Number:	1.2.2.2
Name:	Coil
Part Number:	Copper 303-400-445
Component of:	1.2.2 Coil SA
Component List:	-----
Geometry:	<div> <div>part;</div> <div>rotation;</div> <div>translation;</div> <div>scale</div> <div> <div>↑</div> <div></div> <div></div> <div>↓</div> </div> </div>
Cost:	\$0.50
Manufacturing:	
Manufacturing Cost:	\$0.50
Operation:	
MTBF:	<model output>
Operation Task List:	-----
Operation Cost:	SUM(All Operation Time * Work Code RATE)
Maintenance:	
MTTM:	-----
Maintenance Task List:	-----
Maintenance Cost:	SUM(All Maintenance Time * Work Code RATE)
Repair:	
MTTR:	-----
Repair Task List:	-----
Repair Cost:	SUM(All Repair Time * Work Code RATE)

Figure 4 - Example Assembly List Frame

For assembly lists which contain defined manufacturing, operation, machining, maintenance, or repair tasks, a double click on the task will pop-up the task list for that particular type of task:

Assembly Tasks:

- Assemble
- Attach
- Crimp
- Fasten

Continue

This function for selecting defined tasks is not connected in Phase I. For Phase II, the user will be able to select a task from this predefined list of task, and place it in the assembly list.

Similar in function is the tools pop-up list. A double click on any defined operation which requires a tool will bring up the defined tools window:

Tools:

- Crimping Tool
- Hammer
- Mallet
- Pliers

Continue

As with the task list, this function is not connected in Phase I. For phase II, this feature will allow the user to select a tool from a predefined set of tools, and then place that tool in the assembly list for a given component.

At some points in an assembly list the words

<model output>

highlighted in red will appear. This specifies that the value for this field is derived from a calculation during a run. This field will be filled in, highlighted in red, after the user executes a run by selecting Run under the Options menu. Note that if any of the parameters which affect the model output are changed after a run, the value is replaced with the <model output> indicator.

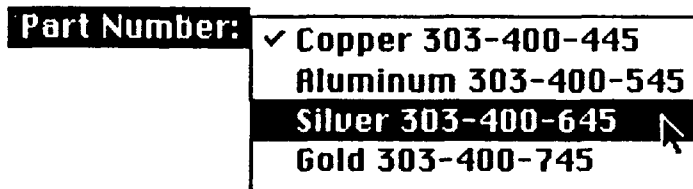
Values in the assembly list which are highlighted in blue are data base values which are automatically set when the user changes the thickness, current, spring, coil, or plating material.

Some assembly lists contain more than one screen worth of information. To change screens, simply click the mouse in the up or down page arrow. In Phase II, all Assembly List information will be on a single, scrollable screen.

There are a total of five model parameters in three assembly lists which can be changed by the user as follows:

<u>Assembly List</u>	<u>Item</u>
Coil	Coil Material
Spring	Part Number
Blade	Plating Material
"	Plating Thickness (in inches)
"	Current (in mAmps)

The coil material, spring part number, and blade plating material are changed by clicking on the field which invokes a pop-up window of the available choices. The user simply drags the pointer to the desired value in the pop-up and then releases the mouse:



All relevant data for the selection made is automatically loaded from a data base.

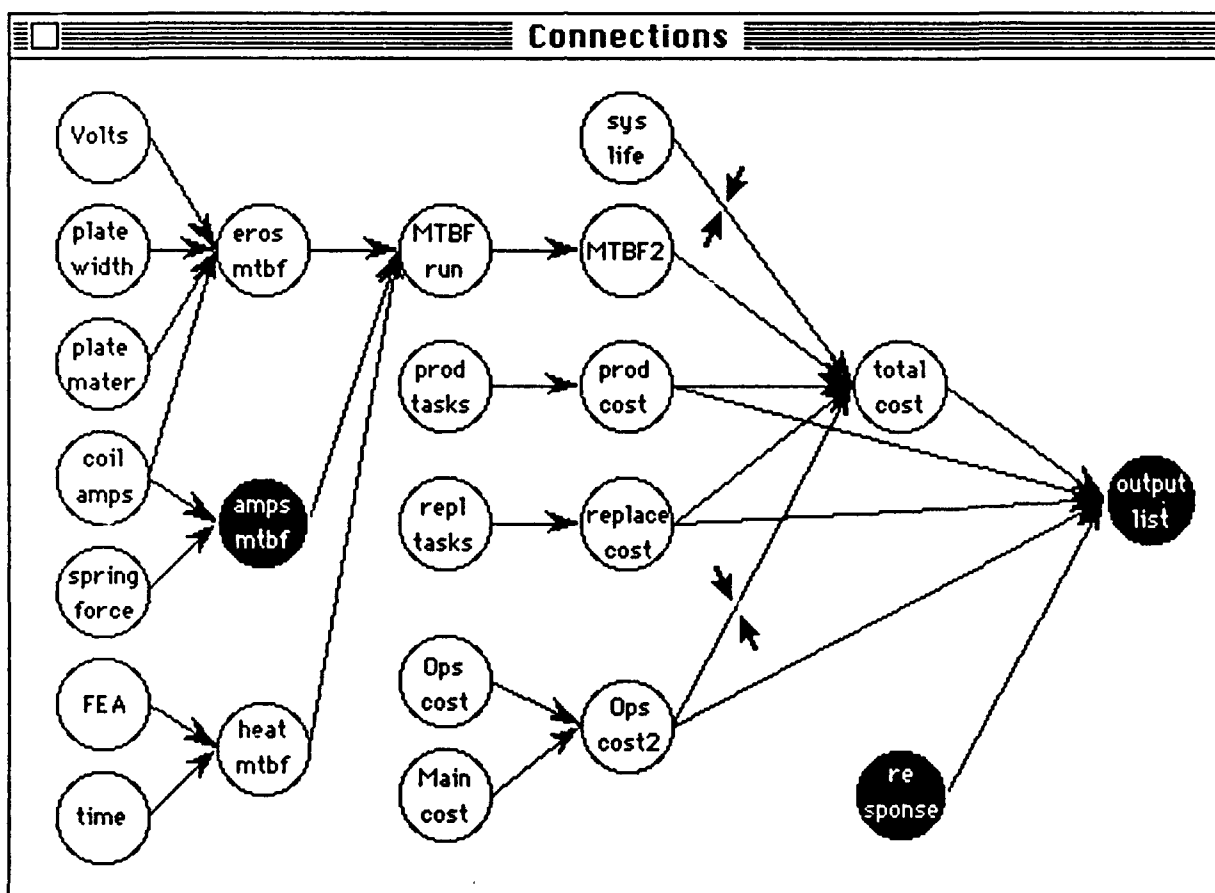
The plating thickness and the current are changed by clicking the mouse in the text edit box, entering the desired number for the current or the thickness, and then pressing the <return> key.



To ensure that the value entered for the blade thickness or the current is accepted, always exit the second blade assembly list screen by clicking in the page up arrow!!

Connections -

This option is used to view the flow of the inputs and outputs for the user defined models. For this Phase I proof-of-concept program, this function is not activated. In the Phase II ULCE program, this function will allow the user to specify the field and format for model input and output. The integration of different models will be accomplished by the use of Glue, a software system for model integration, which is a KDT software product for integrating diverse models in this fashion. An example Connections Layer is seen below:

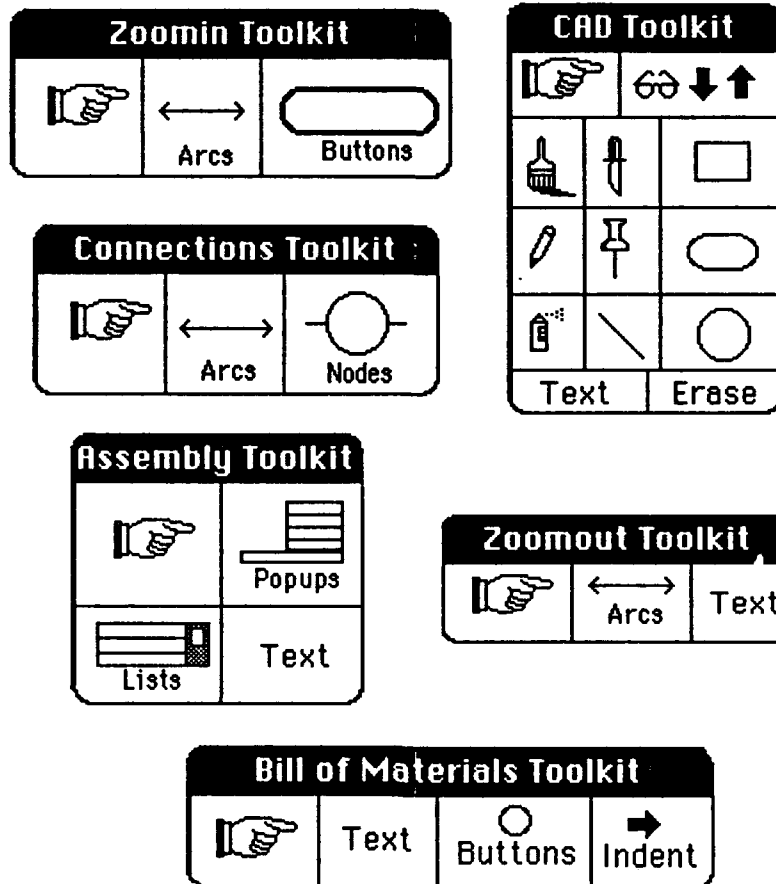


Three of the nodes, highlighted, and two of the arcs, noted by pointer arrows, are connected to further information and example screens by a single click of the mouse. The user is encouraged to explore these definitions and examples while viewing the connections layer.

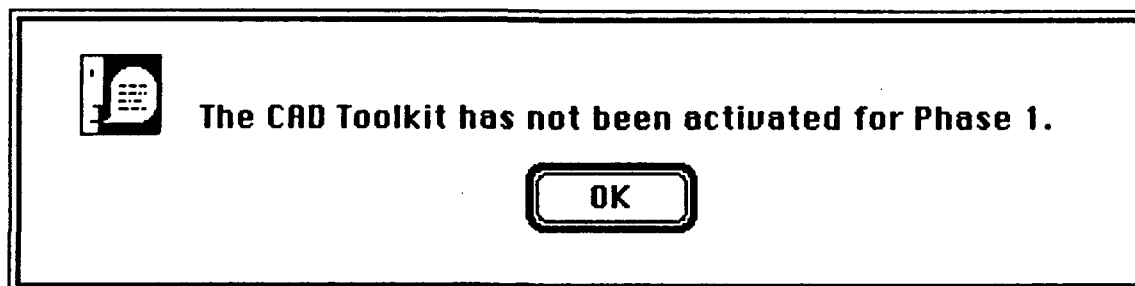
FEA -

Not implemented in Phase I. In Phase II, this option will display the results of a finite element analysis of the selected active component.

Toolkits - Each layer has its own toolkit for creating the fields and defining the objects within the layer as depicted below:



These toolkits are not connected for Phase I, and none of the fields or objects in the pre-defined layers may be changed. In Phase II, the toolkit will be a second main window which will have a constant focus. For this Phase I proof-of-concept program, there is only one main focus window. However, the focus window may be changed by simply moving the mouse pointer to the desired window. By clicking the mouse and holding it down in the black title region of the toolbox, the toolbox window may be moved anywhere on the screen. Attempting to select an item in the toolkit will produce an error dialog similar to the one seen below:



Sample ULCE Session

The following is a step by step guide of a short but representative ULCE session:

- 1 Enter ULCE program via a double click on the ULCE icon.
- 2 Click once on the "About ULCE" window.
- 3 Click once on the "Programmed By" window.
- 4 Pull down the "File" menu and select "Open Saved Design."
- 5 Select "default_ULCE" from the "ULCE folder" and click on "Open."
- 6 Get an overview of the entire system - Pull down the "Layer" menu, select "Parts Tree", and select "Zoomed Out."
- 7 Obtain the default model outputs - Pull down the "Options" menu and select "Run."

For this example case, the first goal is to increase the MTBF

- 8 Pull down the "Options" menu, select "Advice On", and select "MTBF."
- 9 Select "Erosion MTBF" and double click the mouse.
- 10 Select "Blade Plating Thickness" and double click the mouse.
- 11 Note the current thickness is .01. Click in the "Advice Thickness" text box and change the thickness to 0.02. Press the <Return> key.

Note that this increases the erosion MTBF, which is what was desired. Now change the plating thickness in the ULCE session model.

- 12 Click in the "Done" button.
- 13 Pull down the "Layer" menu and select "Bill of Materials."
- 14 Click in the radio button next to "1.2.1.3.2 Blade." This makes "Blade" the active component.
- 15 Take a look at the blade. Pull down the "Layer" menu and select "CAD Drawing."
- 16 Pull down the "Layer" menu and select "Assembly List."
- 17 Click in the "Page" down arrow.
- 18 Click in the "Plating Thickness" box and change the thickness to 0.02. Press the <Return> key.
- 19 Click in the "Page" up arrow.

Note that the blade cost (highlighted in blue) has increased due to the change in thickness. Also note that the plating time (next to last item in the machining task list) has increased.

- 20 Pull down the "Options" menu and select "Run."
- 21 Compare this with the default values. Pull down the "Options" menu and select "History."

Note that the MTBF increased from 157.7 to 169.2 hours (7% increase), thus the total cost of the system was reduced. However, also note the the production cost of single switch rose from \$49.96 to \$62.46 (25% increase!!). The second goal of this example will be to attempt to decrease the production cost.

- 22 Pull down the "Options" menu, select "Advice On", and select "Cost."
- 23 Select "Manufacturing Cost" and double click the mouse.
- 24 Select "Assembly Cost" and double click the mouse.
- 25 Select "Coil Cost" and double click the mouse.
- 26 Click on the "Advice Coil" and pull down to select "Aluminum."

Changing the coil material from Copper to Aluminum decreases the cost, which is what was desired. However, note that it also both increases and decreases the MTBF. The best way to determine whether the change from Copper to Aluminum is best in the overall design is to execute the models with an Aluminum coil.

- 27 Click in the "Done" button.
- 28 Pull down the "Layer" menu and select "Zoomed in."
- 29 Click in the "Coil" button.
- 30 Pull down the "Layer" menu and select "Assembly List."
- 31 Click on "Part Number" and pull down to select "Aluminum 303-400-545."
- 32 Pull down the "Options" menu and select "Run."
- 33 Compare this with the previous execution. Pull down the "Options" menu and select "History."

Changing the coil material to Aluminum did indeed reduce the price for each individual switch, but only by 20 cents. Note that the MTBF was lower, thus the entire cost of the system was increased slightly. The biggest penalty was paid in the response time which is now 3% slower.

This process can be repeated many times until a best combination of price, MTBF, and response time is found.